

**T.C.**  
**TRAKYA ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**MPEG-2 KODLANMIŞ**  
**VIDEO GÖRÜNTÜLERİNİN**  
**İÇERİK TABANLI SORGULANMASI**

**CEM TAŞKIN**  
**DOKTORA TEZİ**

**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**  
**DANIŞMAN: YRD.DOÇ DR. NURŞEN SUÇSUZ**

**EDİRNE, 2010**

**T.C.**  
**TRAKYA ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**MPEG-2 KODLANMIŞ VIDEO GÖRÜNTÜLERİNİN**  
**İÇERİK TABANLI SORGULANMASI**

**Cem TAŞKIN**

**Doktora Tezi**

**Bilgisayar Mühendisliği Anabilim Dalı**

Bu tez 30/03/2010 tarihinde aşağıdaki jüri tarafından kabul edilmiştir.

**Prof.Dr. Mesut RAZBONYALI**  
Başkan

**Yrd.Doç.Dr. Nurşen SUÇSUZ**  
Danışman

**Yrd. Doç.Dr.Aydın CARUS**  
Üye

**Yrd.Doç.Dr. Deniz TAŞKIN**  
Üye

**Yrd. Doç.Dr.Hilmi KUŞÇU**  
Üye

## **ÖZET**

Doktora tezi, MPEG-2 Kodlanmış Video Görüntülerinin İçerik Tabanlı Sorgulanması, T.C. Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı.

Bu tezin amacı, günümüzde yaygın olarak kullanılan video sıkıştırma yöntemlerinden olan MPEG-2 yöntemi kullanılarak sıkıştırılmış video görüntülerinin içerik tabanlı sorgulanmasına imkân sağlayan bir sistem geliştirmektir. Geliştirilen sistem MPEG-2 yöntemi kullanılarak sıkıştırılmış video dosyaları içerisinde yer alan resim çerçevelerini elde ederek, bu resim çerçeveleri içerisinde yer alan sahne metni ve yapay metinlerin konumlarını tespit etmektedir. Konumları tespit edilen bu ifadeler, OCR işlemine tabi tutularak, resim çerçevesinin içeriği hakkında bilgiler vermektedir. Elde edilen bu bilgiler veritabanına kayıt edilerek, video dosyasının içerik tabanlı sorgulanması sağlanmaktadır.

Bu tez 2010 yılında yapılmıştır ve 110 sayfadan oluşmaktadır.

**ANAHTAR KELİMELER:** MPEG-2, Video OCR, Video İndeksleme, I-çerçevesi

**ABSTRACT**

PhD Thesis, Content Based Querying for MPEG-2 Coded Video Files, The Division of Computer Engineering, Physical Sciences Institute, Trakya University.

The purpose of this thesis is developing a new system for querying the files which are coded by a popular video compression method MPEG-2. Developed system extracts picture frames from MPEG-2 coded video files, gets the scene texts and artificial texts on that picture frames. Extracted texts will be processed by OCR. The outputs of OCR process give information about picture frames. By storing the information extracted from picture frames in database, the processed video will be suitable for content querying.

This study was completed in 2010 and consists of 110 pages.

**KEY WORDS:** MPEG-2, Video OCR, Video Indexing, I-frame

**TEŞEKKÜR**

Öncelikle yüksek öğrenim sırasında yardımlarını hiçbir zaman esirgemeyen ve akademisyen olmam için beni teşvik eden danışman hocam Yrd.Doç.Dr. Nurşen SUÇSUZ'a teşekkür ederim.

Ayrıca tez çalışmam sırasında yaptıkları yapıcı eleştiriler ile destek olan Prof. Dr. Mesut RAZBONYALI, Yrd. Doç. Dr. Aydın CARUS ve Yrd.Doç.Dr. Deniz TAŞKIN'a, çalışmalarımda bana sürekli destek olan aileme ve çalışma arkadaşlarıma teşekkürlerimi sunarım.

## İÇİNDEKİLER

<b>ÖZET .....</b>	<b>i</b>
<b>ABSTRACT .....</b>	<b>ii</b>
<b>TEŞEKKÜR.....</b>	<b>iii</b>
<b>ŞEKİLLER LİSTESİ.....</b>	<b>vi</b>
<b>ÇİZELGELER LİSTESİ.....</b>	<b>x</b>
<b>EKLER LİSTESİ.....</b>	<b>xii</b>
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. VİDEO OCR SİSTEMİ.....</b>	<b>4</b>
2.1. Video OCR Sisteminin Bileşenleri.....	6
2.2. Video OCR Sisteminin Veritabanı Tasarımı.....	7
<b>3. VİDEO SIKIŞTIRMA.....</b>	<b>9</b>
3.1. Hareketli Video Görüntülerini Sıkıştırma İhtiyacı .....	10
3.2. MPEG Çalışma Grubu .....	14
3.3. MPEG-2 Video Sıkıştırma Yöntemi.....	15
3.3.1. Çerçeve içi kodlama .....	17
3.3.1.1 RGB renk uzayından YUV renk uzayına dönüşüm.....	17
3.3.1.2 Resim çerçevesinin 8x8 boyutunda bloklara ayrılması.....	21
3.3.1.3 Ayrık kosinüs dönüşümü.....	22
3.3.1.4 Nicelendirme.....	23
3.3.1.5 Zigzag tarama.....	26
3.3.1.6 Değişken uzunlukta kodlama.....	28
3.3.2. Çerçeveler arası kodlama .....	29
<b>4. MPEG-2 VİDEO AKIMI .....</b>	<b>32</b>
4.1. Video Akımı Başlığı .....	34
4.2. Genişletilmiş video akımı başlığı .....	37
4.3. Resim Grubu başlığı .....	39
4.4. Resim Başlığı .....	41
4.5. Genişletilmiş Resim Başlığı.....	43
4.6. Dilim Başlığı .....	45
4.7. Makro Blok Kodlama .....	47
4.7.1. Makro blok tipi belirle() alt yordamı.....	49

4.8. Blok kodlama() yordamı .....	51
<b>5. MPEG-2 YÖNTEMİ KULLANILARAK SIKIŞTIRILMIŞ DOSYA İÇERSİNDEN I-ÇERÇEVELERİNİN ELDE EDİLMESİ .....</b>	<b>55</b>
5.1. Sıkıştırılmış dosya içersinde yer alan I-çerçevesinin tespiti.....	56
5.2. I-çerçevesinin oluşturan dilimlerin tespiti.....	60
5.3. Makro blok kodlama.....	63
5.4. Blok kodlama .....	64
5.5. Video çözme işlemleri .....	69
<b>6. VIDEO OCR SİSTEMLERİ ÜZERİNE YAPILAN ÇALIŞMALAR.....</b>	<b>76</b>
6.1. Sıkıştırılmış Video Görüntüsünü Çözerek Elde Edilen Resim Çerçevesi İçersinden Metin İfadeyi Elde Eden Yöntemler .....	76
6.2. Sıkıştırılmış Dosya İçersinden Metin İfadeleri Tespit Eden Yöntemler .....	81
<b>7. VIDEO OCR SİSTEMİNDE METİN İFADELERİ TESPİT ETMEK İÇİN ÖNERİLEN YÖNTEM .....</b>	<b>84</b>
7.1. Bağlı Bileşen Analizi .....	87
7.2. Resmin Sınır Çizgilerinden Metin İfadeleri Elde Etme.....	89
7.3. Metin ifade olabilecek parçaların birleştirilmesi.....	92
<b>8. GELİŞTİRİLEN VIDEO OCR SİSTEMİ.....</b>	<b>94</b>
<b>9. SONUÇ .....</b>	<b>98</b>
<b>KAYNAKLAR.....</b>	<b>99</b>
<b>EKLER.....</b>	<b>102</b>
<b>ÖZGEÇMİŞ.....</b>	<b>111</b>

## ŞEKİLLER LİSTESİ

Şekil 2.1. Örnek Resim Çerçevesi .....	5
Şekil 2.2. Video OCR Sistemi.....	6
Şekil 2.3. Video OCR sistemimin veritabanı tasarımı.....	8
Şekil 3.1. Video Görüntüsü Çerçevelerden Oluşmaktadır .....	9
Şekil 3.2. Resim çerçevesi ve pikseller.....	10
Şekil 3.3. 8 Bit,16 Bit ve 24 Bit renk derinlikleri için renk paleti.....	11
Şekil 3.4. Video Çerçeveleri .....	15
Şekil 3.5. MPEG-2 sıkıştırma yönteminde kullanılan çerçeve tipleri .....	16
Şekil 3.6. RGB renk uzayında yer alan ana renkler.....	18
Şekil 3.7. Blok ve makro bloklar .....	19
Şekil 3.8. 4:2:0 Renk bilgisi alt örnekleme .....	20
Şekil 3.9. Resim çerçevesinin dilim, makro blok ve bloklara bölünmesi.....	22
Şekil 3.10. Ayrık kosinüs dönüşümü .....	23
Şekil 3.11. Standart nicelendirme matrisleri .....	24
Şekil 3.12. Örnek nicelendirme matrisi .....	25
Şekil 3.13. Nicelendirme işlemi sonucunda elde edilen matris.....	25
Şekil 3.14. Zigzag taramada kullanılan tarama desenleri .....	26
Şekil 3.15. Zigzag tarama adımları.....	27
Şekil 3.16. Çerçeve içi kodlamanın aşamaları .....	29



Şekil 3.17. Çerçeveler arası kodlama.....	30
Şekil 4.1.Başlangıç kodu yapısı.....	33
Şekil 4.2. Video akımı başlangıç kodu .....	35
Şekil 4.3. Genişletilmiş video akımı başlangıç kodu.....	38
Şekil 4.4. Resim grubu başlangıç kodu.....	39
Şekil 4.5. Resim başlığı.....	41
Şekil 4.6. Dilimlere bölünmüş örnek resim çerçevesi .....	45
Şekil 4.7. Dilim başlangıç kodu .....	45
Şekil 4.8. Parlaklık bilgisi içeren bloklar .....	51
Şekil 5.1. MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya yapısı.....	55
Şekil 5.2. Resim Grubu .....	56
Şekil 5.3. MPEG-2 yöntemiyle sıkıştırılmış dosyalar üzerinde başlık bilgisi bulan.....	57
Şekil 5.4. MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içerisinde başlık bilgilerinin konumunu ve sayısını bulan örnek program .....	58
Şekil 5.5. Çerçeve tiplerini ve çerçeve geçici referans numaralarını bulan .....	59
Şekil 5.6. Dosya içerisindeki konumu verilmiş olan resim çerçevesi ile bilgi döndüren fonksiyon.....	60
Şekil 5.7. Örnek Resim Çerçevesi .....	61
Şekil 5.8. Resim çerçevelerini oluşturan dilimlerin başlangıç ve bitiş konumlarını tespit eden C# kodu.....	62
Şekil 5.9. Dilimleri kodlamak için kullanılan C# kodu .....	63

Şekil 5.10. Örnek resim çerçevesinin 1. Dilimini kodlayan bit dizisi .....	67
Şekil 5.11. Örnek resim çerçevesine ait 1.dilimin çözülme aşamaları –Dilim başlığı ve makro blok kodlama .....	68
Şekil 5.12. Örnek resim çerçevesine ait 1.dilimin çözülme aşamaları – Blok kodlama.	69
Şekil 5.13. Ters tarama işlemi .....	70
Şekil 5.14. DC ve AC katsayılarını elde etmek için kullanılan formül .....	73
Şekil 5.15. Doymuluk ve yanlış eşleştirme kontrolleri.....	74
Şekil 6.1. Algoritmanın blok diyagramı.....	77
Şekil 6.2. Sınıflandırma yapan yapay sinir ağının yapısı.....	80
Şekil 6.3. Sıkıştırılmış video dosyası içerisinde yer alan metin ifadelerin yerini tespit eden algoritma .....	82
Şekil 7.1. Örnek resim .....	85
Şekil 7.2. Örnek resim içerisinde yer alan bir karakter.....	85
Şekil 7.3. Örnek Uygulama .....	86
Şekil 7.4. Normalizasyon işlemi yapan fonksiyon .....	87
Şekil 7.5.Bağlı bileşenler analizinin tüm resim üzerinde uygulanması.....	87
Şekil 7.6. Referans Piksel ve Komşu Pikseller .....	88
Şekil 7.7. checkPixel() fonksiyonu .....	88
Şekil 7.8. Resim içerisinde yer alan nesnelerin sınır çizgilerinin belirlenmesi.....	89
Şekil 7.9. Sonuç resmin yakınlaştırılmış hali.....	90

Şekil 7.10. findCC ve savePart fonksiyonlarının kullanımı.....	91
Şekil 7.11. Resim Parçaları .....	91
Şekil 7.12. Birleştirilmiş Resim Parçaları.....	92
Şekil 7.13. Birleştirilmiş Resim parçalarının ters dönüştürülmüş halleri .....	93
Şekil 8.1. Örnek video dosyası içerisinde elde edilmiş I-çerçevesi.....	94
Şekil 8.2. Örnek resim çerçevesi üzerinde önerilen yöntemin uygulanması sonucunda elde edilen resim parçaları .....	95
Şekil 8.3. Geliştirilen Video OCR sisteminin ara yüzü .....	95
Şekil 8.4. Örnek video görüntüsü üzerinden elde edilmiş olan I-çerçevelerinin bir bölümü .....	96
Şekil 8.5. Arama işlemi için kullanılan ara yüz .....	97

## ÇİZELGELER LİSTESİ

Çizelge 3.1. Renk derinliği tablosu.....	11
Çizelge 3.2. Video Standartları .....	12
Çizelge 3.3. MPEG grubunun çalışmaları .....	14
Çizelge 3.4. RGB renk uzayından örnek renkler ve R,G,B yoğunluk değerleri .....	18
Çizelge 4.1. MPEG-2 yönteminde kullanılan başlangıç kodları .....	33
Çizelge 4.2. Video akımı başlığı .....	36
Çizelge 4.3. Bir saniyede işlenecek olan çerçeve sayısı tablosu .....	37
Çizelge 4.4. Genişletilmiş video akımı başlığı.....	38
Çizelge 4.5. Renk bilgisi alt örnekleme formatı çizelgesi .....	39
Çizelge 4.6. Resim grubu başlığı.....	40
Çizelge 4.7. Zaman kodu .....	40
Çizelge 4.8. Resim başlığı.....	42
Çizelge 4.9. Çerçeve tipi kodlama tablosu.....	43
Çizelge 4.10. Genişletilmiş Resim Başlığı.....	44
Çizelge 4.11. Dilim başlığı.....	46
Çizelge 4.12. Makro blok kodlama.....	47
Çizelge 4.13. Makro blok adres artım değeri için değişken uzunlukta kodlama tablosu	49
Çizelge 4.14. Blok kodlama() yordamı.....	52

Çizelge 4.15. Parlaklık bilgisi içeren bloklar kodlanırken DC katsayısı farkını kodlamakta kullanılacak bit sayısı.....	53
Çizelge 4.16. Renk bilgisi içeren bloklar kodlanırken DC katsayısı farkını kodlamakta kullanılacak bit sayısı.....	53
Çizelge 5.1. Değişken uzunlukta kodlama için kullanılacak tabloların seçimi.....	65
Çizelge 5.2. Nicelendirme ölçüt değeri tablosu.....	72
Çizelge 5.3. Çerçeve içi kodlama DC duyarlılık değeri - DC katsayısı çarpanı değeri tablosu .....	73

**EKLER LİSTESİ**

EK-A. Tablo B-14.....	103
EK-B. Tablo B-15.....	108
EK-C. Kısaltmalar Listesi.....	112

## 1. GİRİŞ

Günümüzde bilgisayar sistemleri her alanda yaygın olarak kullanılmaktadır. İnternet ağının hızında yaşanan gelişmelere paralel olarak, paylaşılan bilgi türleri de farklılık göstermektedir. Başlangıçta sadece metin dosyaların etkin biçimde paylaşılmasına imkân veren internet ağı, bugün anlık ses ve görüntü paylaşımının yapılmasına da olanak sağlamaktadır.

Günden güne sayısal ortamda paylaşılan, depolanan, çoklu ortam verisi artış göstermektedir. Bu artışa paralel olarak da çoklu ortam verilerinin içeriğine bakılarak sorgulama yapabilen bilgisayar sistemlerine ihtiyaç duyulmaktadır. Bu ihtiyacı karşılamak için içerik tabanlı sorgulama üzerine birçok çalışma yapılmaktadır. Çalışmalar başlangıçta resimler üzerindeki nesnelerin içeriğini sorgulamaya yöneliktir. Bir resmin içerisinde yer alan nesnelerin tespit edilmesi, nesnelerin birbirlerine göre konumlarının sorgulanması, resmin içerisinde yer alan metin ifadelerin tespiti ve resmin içeriği hakkında sorgulama yapmaya yönelik çalışmalar yapılmış ve olumlu sonuçlar alınmıştır. Özellikle resim içerisinde yer alan metin ifadeler resmin içeriği hakkında kesin bilgiler elde etmeye yardımcı olmaktadır.

Sadece resimlerin değil aynı zamanda hareketli video görüntülerini içeren çoklu ortam verilerinin de sayısal ortamda depolanması, araştırmacıları bu verileri sorgulama ihtiyacına yönelik çalışmalara yönlendirmiştir. Resim ile karşılaştırıldığında oldukça fazla depolama alanına ihtiyaç duyan hareketli video görüntüleri içerik tabanlı sorgulamak için resim üzerinde kullanılan yöntemlerin kullanılması mümkün olmamaktadır. Bu yüzden hareketli video görüntüleri üzerinde sorgulama yapan yöntemler ile resim üzerine sorgulama yapan yöntemler farklılık göstermektedir.

Sayısal ortamda depolanan hareketli video görüntülerinin içerik tabanlı sorgulanması kısa zaman içerisinde yaygınlaşması kaçınılmaz olan IPTV ile birlikte daha da önem taşıyacak ve yaygın kullanım sahası bulacaktır. Hareketli video görüntüleri, resim dosyalarına göre oldukça fazla depolama sahasına ihtiyaç duymaktadır. Bu ihtiyaç hareketli video görüntüsünün sayısal ortamda depolanma maliyetini arttırmaktadır. Bu kaçınılmaz maliyeti minimize etmek amacıyla hareketli

video dosyalarının sayısal ortamda etkin bir biçimde sıkıştırılmasını sağlamak için 1988 yılında MPEG adlı uluslar arası bir çalışma grubu kurulmuştur. Bu çalışma grubu 20 farklı ülkeden 300 uzman katılımcıdan oluşmuş ve ISO tarafından standart olarak kabul edilmiş birçok başarılı çalışmaya imza atmıştır. Oluşturulan MPEG-1 ve MPEG-2 standartları endüstri firmaları tarafından da kabul görmüş ve halen günümüzde de aktif olarak kullanılmaktadır. Birçok platformda çalışabilen MPEG kodlayıcılar ve çözücüler geliştirilmiştir.

Kabul görmüş olan MPEG-2 standardı günümüzde bilgisayar ortamında, MPEG kod çözücüyeye sahip birçok cihazda ve IPTV’de de kullanılan bir formattır. MPEG-2 yöntemi yüksek bir sıkıştırma oranına sahiptir. Tezin üçüncü ve dördüncü bölümlerinde bu yöntem detaylı olarak anlatılacaktır. Tezin beşinci bölümünde, MPEG-2 yöntemi kullanılarak sıkıştırılmış olan hareketli video görüntü dosyası içersinden, resim çerçevelerinin elde edilmesi ayrıntılı olarak anlatılacaktır. Sahne metni ve yapay metin tespiti için kaynak olarak elde edilen bu resim çerçeveleri kullanılacaktır. MPEG-2 kullanılarak sıkıştırılmış hareketli video görüntüsü içersinden resim çerçevelerinin elde edilmesi için ISO tarafından yayınlanmış olan ISO/IEC 13818-1 ve ISO/IEC 13818-2 numaralı belgelerden faydalanılacaktır.

Resim çerçeveleri elde edildikten sonra, Video OCR sisteminin performansını önemli ölçüde etkileyen, resim çerçeveleri içersinde yer alan sahne metni ve yapay metinleri tespit eden bileşene işletilmek üzere aktarılmaktadır. Tezin altıncı bölümünde, resimler içersinde yer alan metin ifadelerin tespiti ile ilgili çalışmalar ayrıntılı incelenerek, tezin yedinci bölümünde resim çerçeveleri içersinde yer alan sahne metni ve yapay metinleri tespit etmek için kullanılabilir yeni bir yöntem sunulmaktadır.

Bu tez çalışmasında özellikle MPEG-2 standardına göre sıkıştırılmış olan video görüntülerinin içerik tabanlı sorgulamasını yapabilen bir sistem önerilmektedir. Hareketli video görüntüsü, belirli bir zaman aralığında birbiri ardına gösterilen resim çerçevelerinden oluşmaktadır. Bu resim çerçevelerinde, video görüntüsünün içeriği hakkında fikir yürütülmesine yardımcı olabilecek metin ifadeler bulunabilir. Bu metin ifadeler sahnenin içersinde yer alan objelerin üzerinde bulunan doğal ifadeler olabileceği gibi (sahne içersinde yer alan bir eşyanın markası gibi), özel olarak sonradan



eklenmiş yapay bilgiler (bir televizyon dizisinde sađ alt köşede yer alan dizinin ismi) de olabilir. Önerilen sistem, MPEG-2 yöntemi kullanılarak sıkıştırılmış hareketli video görüntüleri üzerinde içerik tabanlı sorgulama yapabilmektedir.

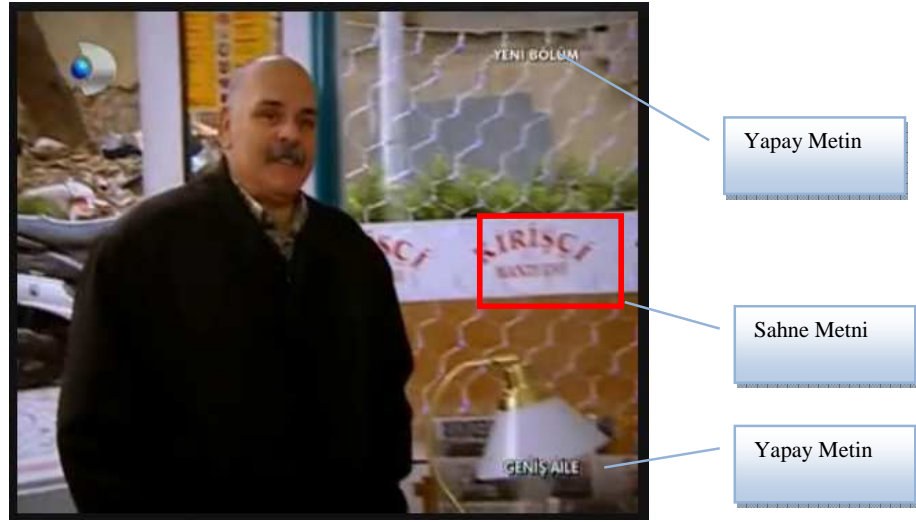
## 2. VIDEO OCR SİSTEMİ

Video OCR sistemi, hareketli video görüntüleri içerisinde yer alan metin ifadeleri tespit edebilen sistemdir. Hareketli video görüntüsü belirli bir zaman aralığında birbiri ardına gösterilen resim çerçevelerinden oluşmaktadır. Sayısal olarak depolanan hareketli video görüntü miktarının artmasıyla birlikte, depolanan bu görüntüleri arşivlemek, gerektiğinde de sorgulamak için Video OCR sistemleri kullanılmaktadır. Günümüzde yeni kullanılmaya başlanan IPTV, sayısal olarak depolanan görüntüler üzerinde sorgulama yapabilen sistemlere sık sık ihtiyaç duyacaktır.

Video OCR sisteminin işlevi, hareketli görüntüyü oluşturan resim çerçeveleri içerisinde yer alan metin ifadeleri tespit etmektir. Hareketli görüntü içerisinde yer alan metin ifadelerin yazıtipi, boyutu ve rengi standart değildir. Ayrıca çoğu zaman gündelik hayattan kayıt edilmiş olan hareketli video görüntüleri, içerisinde birden fazla rengi barındıran karmaşık arka planlar içerebilmektedir. Bu durumda, resim çerçevesi içerisinde yer alan metin ifadeyi arka plandan ayırt edebilmek oldukça güç olmaktadır. Klasik karakter tanıma yöntemlerini kullanarak bu metin ifadelerin karmaşık arka plandan ayırt edilerek tanınması mümkün değildir. Buna karşın hareketli video görüntülerini oluşturan resim çerçeveleri içerisinde yer alan metin ifadelerin bazı karakteristik özellikleri, bu ifadelerin tespit edilmesini kolaylaştırmaktadır. İşte bu avantaj ve dezavantaj olarak sayılabilecek noktalar dikkate alınarak, klasik karakter tanıma yöntemlerinin dışında özel bir algoritma tasarlanarak hareketli video görüntüsü içerisinde yer alan metin ifadelerin tespit edilmesi mümkündür.

Resim içerisinde yer alan metin ifadeleri temel olarak 2 türe ayırmak mümkündür. Sahne metni ve yapay metin (Lienhart ve Effelsberg, 1998). **Sahne metni**, sahnenin doğal yapısından ötürü hareketli görüntüyü oluşturan resim çerçevesi içerisinde yer alan metin ifadedir. Hareketli görüntüyü oluşturan resim çerçevesi içerisinde yer alan herhangi bir objenin üzerinde yer alan metin ifade sahne metni olarak adlandırılabilir. **Yapay metin** ise, hareketli görüntüyü oluşturan resim çerçeveleri içersine özel olarak ve sonradan eklenmiştir. Genellikle yapay metin ifadeler özel yazılımlar kullanılarak

video içersine eklenmektedir. Kayıt yapabilen kamera cihazlarının birçoğu görüntünün çekildiği tarihi ya da özel bir metin ifadeyi resim çerçevesi içersine dâhil edebilen fonksiyonlara sahiptir. Yabancı dilde dublajı olan sinema filmlerinde gösterilen alt yazılar özel yazılımlarla birer yapay metin ifade olarak resim çerçevelerine eklenmektedir. Şekil 2.1’de içersinde sahne metni ve yapay metin bulunan bir resim görüntülenmektedir.

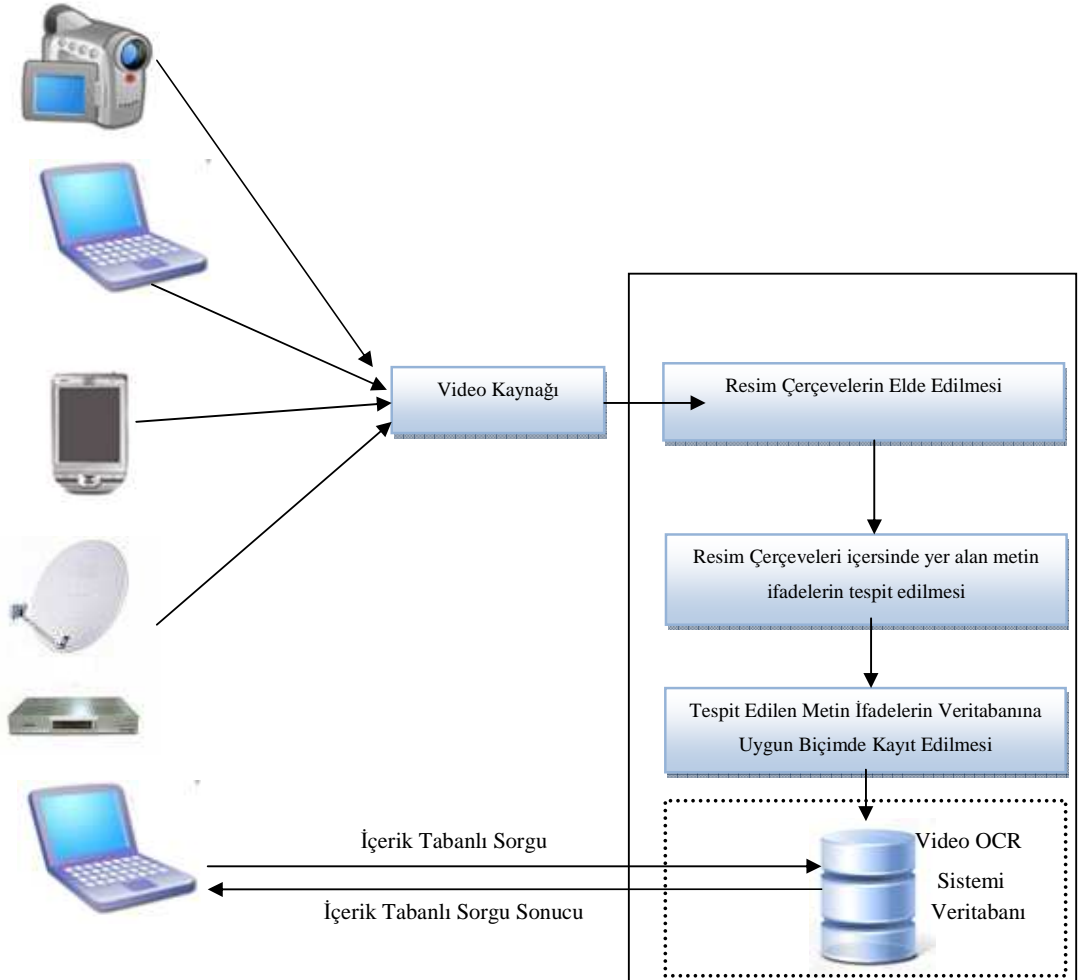


Şekil 2.1. Örnek resim çerçevesi

Video OCR sistemlerinin performansını, hareketli görüntüyü oluşturan resim çerçeveleri içersinde yer alan metin ifadeyi belirlemek için kullanılan algoritmanın performansı önemli ölçüde etkilemektedir. Resim içersinde yer alan sahne metni, o resim çerçevesinin içeriği hakkında kesin bir bilgi içermezken, sonradan eklenen yapay metin o resim çerçevesi hakkında kesin bir bilgi içermektedir. Şekil 2.1’de gösterilen örnekte yapay olarak eklenmiş “YENİ BÖLÜM” ile “GENİŞ AİLE” ifadeleri bu çerçevenin bir TV dizisine ait olduğunu tespit etmek için kullanılabilir kesin bilgilerdir. Bunun dışında resim çerçevesi içersinde yer alan sahne metninden elde edilebilecek olan “KİRİŞÇİ” kelimesi ise bu sahne hakkında yorum yapmak için kullanılabilir kesin bir bilgi içeren ifade değildir.

## 2.1. Video OCR Sisteminin Bileşenleri

Video OCR sistemi, birden fazla bileşenden oluşmaktadır. Video görüntülerinin içerisinde yer alan metin ifadelerin tespiti yardımıyla içerik tabanlı sorgulanabilmesi için hareketli görüntüyü oluşturan çerçevelerin kaynaktan elde edilmesi, elde edilen çerçevelerin içerisinde yer alan metin ifadelerin tespit edilmesi ve daha sonra bir veritabanına kayıt edilmesi gerekmektedir. Şekil 2.2’de Video OCR sisteminin bileşenleri görülmektedir. Temel olarak üç ana bileşenden oluşan Video OCR sisteminin her bir ana bileşeni içinde de alt bileşenler bulunmaktadır.



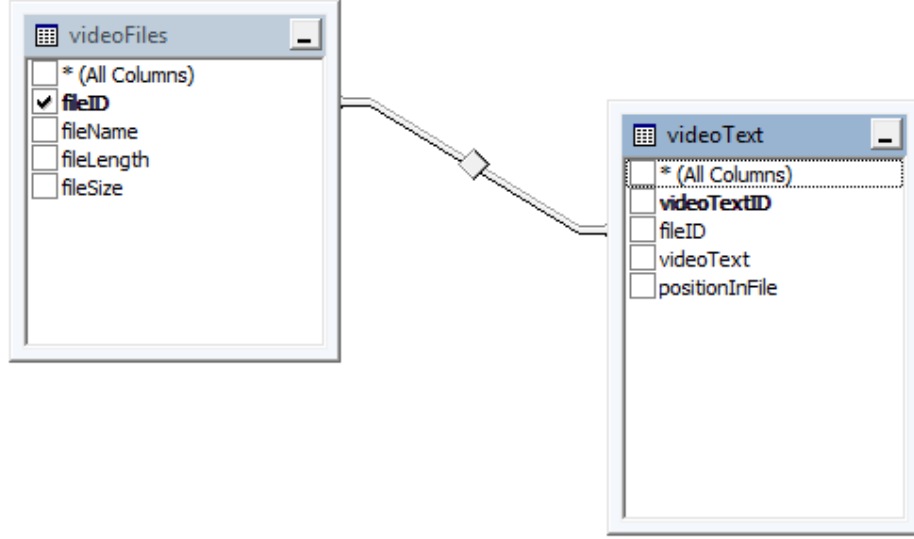
Şekil 2.2. Video OCR Sistemi

Video OCR sisteminde video kaynağı olarak birden fazla seçenek bulunabilir. Bir kamera, bilgisayar, PDA ya da dijital uydu alıcıdan elde edilmiş olan hareketli görüntü, işlemek üzere kaynak olarak Video OCR sistemine sunulabilir. Kaynak tarafından sunulan hareketli video görüntüsünün biçiminin Video OCR sistemi tarafından desteklenmesi gerekmektedir.

Kaynak olarak Video OCR sistemi tarafından desteklenen türde bir görüntü sisteme sunulduğunda, ilk olarak resim çerçevelerine ayrıştırılması gerekmektedir. Eğer kaynak olarak işletilmek üzere sunulan hareketli video görüntüsü sıkıştırılmış ise Video OCR sistemi tarafından tamamı açılmadan da işletebilmesi mümkün olmaktadır. Video dosyaları üzerinde yapılan sıkıştırma işlemi, belirli bir süre aralığında birbiri ardına gelen resim çerçevelerindeki farklılıklarının kodlanması mantığına dayanmaktadır. Bu tez çalışmasında birçok platformda kabul görmüş, yaygın kullanıma sahip MPEG-2 yöntemi kullanılarak sıkıştırılmış dosyaları işleyebilen bir sistem önerilmektedir.

## **2.2. Video OCR Sisteminin Veritabanı Tasarımı**

Video OCR sistemlerinin önemli bir bileşeni sorgulama işlemini gerçekleştiren veritabanı kısmıdır. Sorgulamada esas oluşturacak ölçüt, hareketli video görüntüsü oluşturan resim çerçeveleri içerisinde yer alan sahne metinleri ve yapay metinlerden elde edilecek olan metin ifadelerdir. Önerilen sistem MPEG-2 yöntemi kullanılarak sıkıştırılmış hareketli video görüntüsü içerisinde yer alan I-çerçeveleri üzerinden işlem yapmaktadır. Sistem, veritabanına sahne metni ve yapay metnin bulunduğu her bir I-çerçevenin sıkıştırılmış dosya içerisinde kaçınıcı saniyede görüntülediğini ve çerçeve içerisinden elde edilmiş olan sahne metinlerini ve yapay metinleri kaydetmelidir. Video OCR sistemlerinin etkin bir biçimde kullanıldığı önemli sahalardan biri haber videolarını arşivleyen ve içerik tabanlı sorgulayan sistemlerdir. Haber videoları yapay metin içeriği bakımından oldukça zengindir.

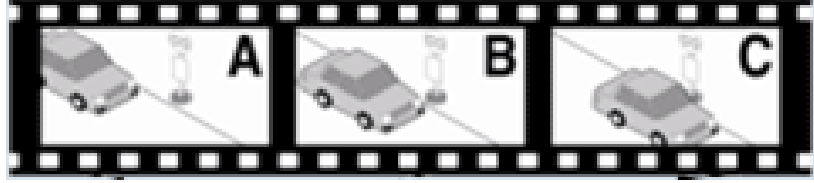


Şekil 2.3. Video OCR sistemimin veritabanı tasarımı

Şekil 2.3'te Video OCR sistemin veritabanı tasarımı görülmektedir. Basit olarak 2 tablodan oluşan tasarımda, işletilmek üzere sisteme gönderilmiş olan her bir dosya **videoFiles** adlı bir tabloya kaydedilmektedir. Bu tabloda yer alan **fileID** sahası birincil anahtar olup, otomatik olarak değeri artan bir sahadır. **videoText** adlı diğer tabloya ise, hareketli video görüntüsünü oluşturan resim çerçevelerinden, içersinden sahne metni ve yapay metinler elde edilmiş olan resim çerçeveleriyle ilgili bilgiler kayıt edilmektedir. Kayıt edilen bilgiler, elde edilen sahne metinleri ve yapay metinlerin dışında, bu resim çerçevesinin hangi dosyaya ait olduğu ve dosyanın içersindeki konumudur (çerçevenin dosya görüntülenirken kaçınıcı saniyede ekranda yer aldığı bilgisi).

### 3. VIDEO SIKIŞTIRMA

Geçmişten günümüze veri sıkıştırma üzerine birçok çalışma yapılmaktadır. Özellikle verilerin sayısal ortamda bir yerden bir yere transfer süresini ve veri depolama maliyetini azaltmak için veri sıkıştırma işlemi yapılmaktadır. Başlangıçta sayısal ortamda sadece metin ifadeler içeren bilgiler depolandığından, metin ifadeler üzerinde sıkıştırma yapan yöntemler geliştirilmiştir. Günümüzde çoklu ortam verileri de bir yerden bir yere iletilmekte ve depolanmaktadır. Metin ifadeler üzerinde sıkıştırma yapan yöntemlerin, hareketli görüntüler üzerinde etkin olarak çalışmadığı görülmüş, ortak bir sayısal görüntü sıkıştırma yöntemini ortaya koyup yaygınlaştırmak amacıyla sayısal kodlanmış hareketli video görüntüleri üzerinde çalışmalar yapmak için 1988 yılında MPEG çalışma grubu kurulmuştur.

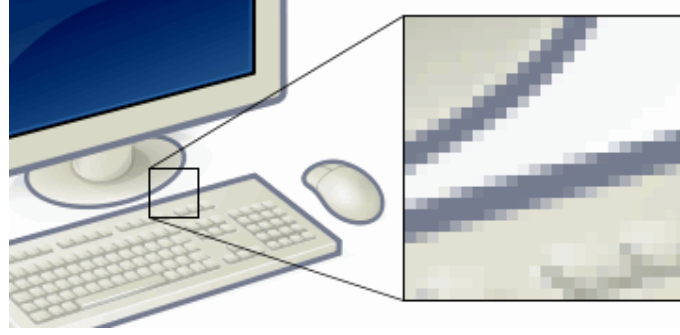


Şekil 3.1. Video görüntüsü çerçevelerden oluşmaktadır

Şekil 3.1'de hareketli video görüntüsünü oluşturan resim çerçeveleri bulunmaktadır. Birbiri ardına gelen bu üç resim çerçevesine bakıldığında sadece belirli kısımlarının değiştiği görülmektedir. Video sıkıştırma yöntemleri, sıkıştırmayı bu kilit noktayı kullanarak yapmaktadır. Bu üç resim çerçevesi için, sadece çerçeveler arasında değişen noktalar kodlanırsa, büyük ölçüde sıkıştırma yapılmaktadır. Sıkıştırılmış görüntü üzerinde açma işlemi yapılırken de sadece resim çerçevelerinin arasındaki farklı noktaların kodlandığı göz önünde bulundurularak, çerçevelerin sıkıştırılmadan önceki görüntüleri elde edilmektedir. Video sıkıştırma yöntemlerinin çoğu bu avantajı kullanarak yüksek oranda sıkıştırma yapmaktadır.

### 3.1. Hareketli Video Görüntülerini Sıkıştırma İhtiyacı

Hareketli video görüntüleri, çok kısa süre aralığında birbiri ardına gösterilen resim çerçevelerinden oluşmaktadır. Resim çerçeveleri ise **piksel** adı verilen noktalardan oluşur. Her bir pikselin koordinatı ve renk değeri bulunmaktadır. Şekil 3.2’de örnek bir resim çerçevesi ve resim çerçevesini oluşturan pikseller görülmektedir.



Şekil 3.2. Resim çerçevesi ve pikseller

Resim çerçevelerini oluşturan piksellerin alabileceği renk değerleri ve bu renk değerlerini bilgisayar ortamında saklamak için piksel başına kullanılacak olan Bit sayısı farklılık göstermektedir. Her bir pikselin renk değerini bilgisayar ortamında saklamak için ihtiyaç duyulan Bit sayısı **renk derinliği** olarak tanımlanmaktadır.

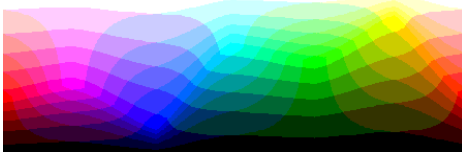
Tek Renkli bir resmi oluşturan piksellerin alabileceği renk değerleri 0 (siyah) ya da 1 (beyaz) olabilir ve Tek Renkli bir resim çerçevesini oluşturan her bir pikselin renk değerini bilgisayar ortamında saklamak için 1 Bite ihtiyaç duyulmaktadır. 8 Bit Gri Ölçekli bir resim çerçevesi oluşturan piksellerin alabileceği renk değeri 0 ile 255 arasındadır ve 8 Bit Gri Ölçekli bir resim çerçevesini oluşturan her bir pikselin renk değerini bilgisayar ortamında saklamak için 8 Bite ihtiyaç duyulmaktadır. Sonuç olarak Tek Renkli bir resim çerçevesi için renk derinliği 1, 8 Bit Gri Ölçekli bir resim çerçevesi için renk derinliği 8’dir. Çizelge 3.1’de bilgisayar ortamında kullanılan renk derinliği tablosu görülmektedir.



Çizelge 3.1. Renk derinliği tablosu

Renk derinliği ismi	Renk derinliği	Farklı renk sayısı
1 Bit - Tek Renkli	1	2
8 Bit - Gri Ölçekli	8	256
8 Bit - Renkli	8	256
15/16 Bit Renkli (Yüksek Renkli)	16	32.768
24 Bit Renkli (Gerçek Renk)	24	16.777.216
30 Bit Renkli (Derin Renk)	30	1.073 Milyar
36 Bit Renkli (Derin Renk)	36	68.71 Milyar
48 Bit Renkli (Derin Renk)	48	281.5 Trilyon

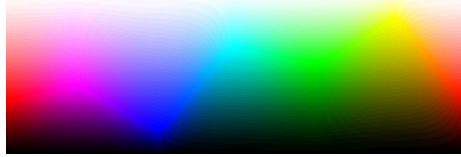
Şekil 3.3’de 8 Bit, 16 Bit ve 24 Bit renk derinlikleri için renk paletleri görülmektedir. Gerçek görüntüye yakın olduğundan, renkli hareketli video görüntüleri için 24 Bit Gerçek Renkli (Gerçek Renk) renk derinliği kullanılmaktadır. Bu yüzden, renkli hareketli video görüntüsünü oluşturan resim çerçevelerinde her bir pikselin renk değeri bilgisayar ortamında 24 Bit yer kaplamaktadır.



a.8 Bit renk derinliği için renk paleti



b.16 Bit renk derinliği için renk paleti



c.24 Bit renk derinliği için renk paleti

Şekil 3.3. 8 Bit,16 Bit ve 24 Bit renk derinlikleri için renk paleti

Resim çerçevesini oluşturan yatay ve dikey piksel sayısı ise **çözünürlük** olarak adlandırılır. Çizelge 3.2’de video standartları ve bu standarda uygun olan video görüntüsünü oluşturan resim çerçevelerinin, çözünürlük değerleri yer almaktadır.

Çizelge 3.2. Video standartları

Video standardı	Çözünürlük
CGA	320 x 200
QVGA	320 x 240
VGA	640 x 480
NTCS	720 x 480
PAL	768 x 576
WVGA	800 x 480
WVGA	854 x 480
WSVGA	1024 x 600
XGA	1024 x 768
HD720	1280 x 768
WXGA	1280 x 768
WXGA	1280 x 800
SXGA	1280 x 1024
WSXGA+	1680 x 1050
SXGA+	1400 x 1050
HD 1080	1920 x 1080
2K	2048 x 1080
WUXGA	1920 x 1200
QXGA	2048 x 1536
WUXGA	1920 x 1200
WQXGA	2560 x 1600
QSXGA	2560 x 2048

Ülkemizde ve çoğu Avrupa ülkesinde TV yayınları PAL video standardı kullanılarak yapılmaktadır. Çizelge 3.2’ye bakıldığında PAL TV standardını destekleyen hareketli video görüntüsünü oluşturan resim çerçevelerinin 768 x 576 piksel boyutunda

olduđu görlmektedir. TV yayında kullanılmak zere kayıt edilen hareketli video grntleri, saniye de 25 ya da 29,97 resim erevesi iermektedir.

Bu noktadan hareketle, bir dakikalık renkli hareketli video grntsnn bilgisayar ortamında sıkıştırılmadan depolanması durumunda ihtiya duyulan depolama alanı hesaplanmak istenirse;

PAL standardı kullanıldığında, her bir resim erevesi

$$768 * 576 = 442.368$$

pikselden oluřmaktadır. Her bir resim erevesinin renk deđerini bilgisayar ortamında saklayabilmek iin 24 Bit gerekmektedir. Bu yzden, PAL standardını destekleyen video grntsne ait bir resim erevesini oluřturan piksellerin renk deđerlerini bilgisayar ortamında depolamak iin;

$$442,386 * 3 = 1.327.104 \text{ Byte} = 1296 \text{ KB} \approx \mathbf{1,265 \text{ MB}}$$

alana ihtiya duyulmaktadır. Hareketli video grntsnn saniyede 25 resim erevesi ierdiđi dřnlrse **1 sn'lik** hareketli video grntsn bilgisayar ortamında sıkıştırılmadan depolamayabilmek iin;

$$25 * 1,265 = \mathbf{31,625 \text{ MB}}$$

alana ihtiya duyulmaktadır. **1 dakikalık** hareketli video grnts iin ise;

$$60 * 31,625 \text{ MB} = \mathbf{1897,5 \text{ MB}}$$

alana ihtiya duyulmaktadır.

Yukarıdaki hesaplamalardan da anlaşılaı gibi, hareketli video grntsnn bilgisayar ortamında sıkıştırılmadan depolanması olduka yksek maliyet getirmektedir. Bu sebeple, bilgisayar ortamında saklanan hareketli video grntsnn hem iletim hem de depolama maliyetini azaltmak iin sıkıştırılarak depolanmaktadır.

### 3.2. MPEG Çalışma Grubu

MPEG çalışma grubunun birçok faydalı çalışması ISO tarafından onaylanmış ve bir standart haline gelmiştir. Birliğin yapmış olduğu çalışmaların listesi Çizelge 3.3’de yer almaktadır.

Çizelge 3.3. MPEG grubunun çalışmaları

Standart	Açıklama	Standart	Yıl
MPEG-1	Video CD ve MP3 standardı	ISO/IEC 11172	1993
MPEG-2	Sayısal Televizyon, Top Box Set ve DVD ‘de kullanılan standart	ISO/IEC 13818	1995
MPEG-4	Ses ve Görüntü objelerini kodlama	ISO/IEC 14496	1999
MPEG-7	Çoklu ortam içeriği tanımlama ara yüzü	ISO/IEC 15938	2002
MPEG-21	Çoklu ortam platformu	ISO/IEC 21000	2001
MPEG-A	Çoklu ortam uygulama formatı	ISO/IEC 23000	2007
MPEG-B	MPEG sistem teknolojileri	ISO/IEC 23001	2006
MPEG-C	MPEG sistem teknolojileri	ISO/IEC 23002	2006
MPEG-D	Çoklu ortam ses teknolojileri	ISO/IEC 23003	2007
MPEG-E	Çoklu ortam platformu	ISO/IEC 23004	2007

Çizelge 3.3’de görüldüğü gibi MPEG, ISO tarafından standart kabul edilmiş ve günümüzde birçok türdeki cihazlar üzerinde kullanılan hareketli görüntü ve ses kodlama biçimini bütünüyle hazırlamıştır. Bu tez çalışmasında önerilen Video OCR sistemi de günümüzde sıklıkla kullanılan MPEG-2 hareketli video görüntüsü ve ses sıkıştırma yöntemini desteklemektedir. IPTV’nin yaygın olarak kullanılmaya başlamasıyla, hareketli görüntülerin içerik tabanlı sorgulama ihtiyacı artacaktır. Birçok IPTV kullanıcısı kendi video arşivlerini yaratacak ve gerektiğinde bu arşiv üzerinde içerik tabanlı sorgulama yapabilecektir.

### 3.3. MPEG-2 Video Sıkıştırma Yöntemi

MPEG-2 video sıkıştırma yöntemi yüksek sıkıştırma oranına sahip hareketli görüntü sıkıştırma yöntemidir. Hareketli video görüntüsü belirli bir zaman aralığında birbiri ardına görüntülenen resim çerçevelerinden oluşmaktadır.

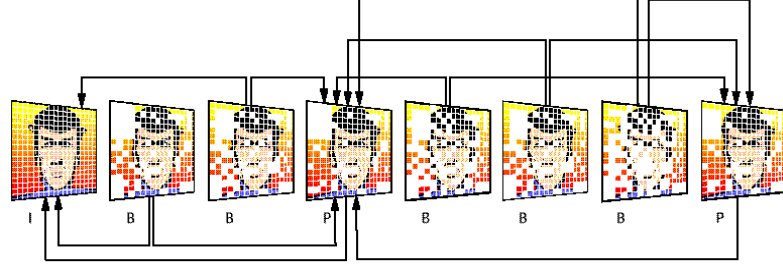


Şekil 3.4. Video çerçeveleri

Şekil 3.4'de hareketli video görüntüsünü oluşturan resim çerçeveleri görülmektedir. Çoğunlukla, birbiri ardına görüntülenen resim çerçeveleri arasında aynı renk değerine sahip pikseller bulunmaktadır. Temel olarak MPEG-2 video sıkıştırma yöntemi hareketli video görüntüsünü oluşturan ve birbirini takip eden resim çerçeveleri arasında tekrar eden piksel değerlerini kodlamayarak sıkıştırma yapmaktadır. Şekil 3.4'de verilen 4 resim çerçevesi için arka plan çok değişiklik göstermemektedir. 4 çerçevede de beyaz kar görüntüsünü oluşturan pikseller tekrar etmektedir. İlk 2 çerçeve için de hareket eden çocuğun görüntüsü çok farklılık göstermemektedir. Bu şekilde, resim çerçeveleri arasındaki farklılıkların belirlenip, sıkıştırılmış dosya içersine sadece bu farklılıkların kaydedilmesi veri tekrarını ciddi oranda azaltmaktadır. Birbirini takip eden resim çerçeveleri arasındaki tekrar eden piksellerin kodlanmayarak yapılan sıkıştırma işlemine **çerçeveler arası kodlama** denir. Bunun dışında MPEG-2 video sıkıştırma yöntemi **çerçeve içi kodlama** olarak adlandırılan bir tekniği de içermektedir. Bu teknik, bir resim sıkıştırma yöntemi olan JPEG'de kullanılan yöntem ile benzerlik göstermektedir.

Çerçeveler arası ve çerçeve içi kodlamanın hangi çerçeveler için yapılacağına, sıkıştırma işlemi yapılırken karar verilmektedir. Sıkıştırılmamış hareketli video görüntüleri için tüm çerçeveler tek tipte olup, her bir çerçevenin tüm piksellerine ait bilgiler dosya içersinde saklanmaktadır. MPEG-2 sıkıştırma yönteminde ise resim

çerçeveleri I-çerçevesi, B-çerçevesi ve P-çerçevesi olmak üzere 3 farklı türdedir. Şekil 3.5’de MPEG-2 sıkıştırma yönteminde kullanılan çerçeve tipleri görülmektedir.



Şekil 3.5. MPEG-2 sıkıştırma yönteminde kullanılan çerçeve tipleri

Çerçeve içi kodlamanın yapıldığı resim çerçeveleri **I-çerçevesi**, çerçeveler arası kodlamanın yapıldığı çerçeveler ise **B-çerçevesi** ve P-çerçevesi olarak adlandırılır. Şekil 3.5’de yer alan çerçeveler incelendiğinde, birçoğunda beyaz kutucukların yer aldığı görülmektedir. Sadece I-çerçevesinde beyaz kutucuklar yer almamaktadır. Beyaz kutucuklar resim çerçevesinin sıkıştırılmış dosya içerisinde kodlanmayan bölümlerini göstermektedir. Bu beyaz kutucuklarda yer alan piksel değerlerine, önceki ya da sonraki resim çerçevelerine bakarak ulaşmak mümkün olmaktadır. MPEG-2 video sıkıştırma yöntemi bu şekilde birbirine yakın resim çerçeveleri arasında ortak olan kısımları kodlamayarak büyük oranda sıkıştırma yapmaktadır. Genellikle, büyük sahne değişimlerinin söz konusu olduğu durumlarda çerçeve içi kodlama yapılırken, diğer durumlarda çerçeveler arası kodlama yapılmaktadır.

I-çerçevesine bakıldığında hiçbir beyaz kutucuğun bulunmadığı görülmektedir. Sıkıştırılmış dosya içerisinde I-çerçevesi oluşturan tüm piksel değerleri kodlanmış olarak yer almaktadır. B-çerçevesinin tam görüntüsünü elde etmek için kendinden önce ve sonra gelen I-çerçevesine ya da P-çerçevesine bakmak gerekmektedir. P-çerçevesinin tam görüntüsünü elde etmek için ise sadece kendinden önce gelen I-çerçevesine ihtiyaç duyulmaktadır. P-çerçevesinde, kendinden bir önce gelen I-çerçevesindeki farklılıklar kodlanmaktadır.

### 3.3.1. Çerçeve içi kodlama

Çerçeve içi kodlama, büyük sahne değişimlerinin söz konusu olduğu resim çerçeveleri üzerinde çalıştırılan bir yöntem olup, çerçeve içi kodlamanın yapıldığı resim çerçeveleri **I-çerçevesi** olarak adlandırılmaktadır. B-çerçevesi ve P-çerçevesinin aksine, I-çerçevesinin tüm piksellerine ait renk değerleri MPEG-2 yöntemi kullanılarak sıkıştırılmış hareketli video dosyası içerisinde yer almaktadır.

I-çerçevelerini oluşturmak için kullanılan çerçeve içi kodlama temelde, başarılı bir görüntü sıkıştırma yöntemi olan JPEG ile benzerlik taşımaktadır. JPEG, ayarlanabilir kayıplı sıkıştırma kullanmaktadır, dolayısıyla JPEG verisinden okunan görüntü ile veriyi yaratmak için kullanılan görüntü aynı değildir. Ancak, kayıplar insan görme sisteminin daha az önem verdiği detaylarda gerçekleştiği için çoğu zaman fark edilmez (Wikipedia).

I-çerçeveleri oluşturulurken kullanılan çerçeve içi kodlama birden fazla aşamadan oluşmaktadır. Bu aşamalar aşağıda listelenmektedir.

1. RGB renk uzayından YUV renk uzayına dönüşüm
2. Resim çerçevesinin 8x8 boyutunda bloklara ayrılması
3. Her bir blok üzerinde
  - a. Ayrık Kosinüs Dönüşümü
  - b. Nicelendirme
  - c. Zig-Zag tarama
  - d. Değişken uzunlukta kodlama

#### 3.3.1.1 RGB renk uzayından YUV renk uzayına dönüşüm

RGB, Red, Green ve Blue kelimelerin kısaltılmış halidir. Şekil 3.6'da RGB renk uzayında yer alan ana renkler görülmektedir. RGB renk uzayında kırmızı, yeşil, mavi ana renkler olarak kabul edilmekte ve diğer tüm renkler bu üç ana rengin belirli

yoğunlukta karıştırılmasıyla elde edilmektedir. Bu üç rengin tam yoğunlukta karışımı ile beyaz renk elde edilirken, sıfır yoğunlukta karışımı ile siyah renk elde edilmektedir.



Şekil 3.6. RGB renk uzayında yer alan ana renkler

1988 yılında VGA teknolojisinin kabul görmesiyle birlikte RGB renk uzayı bilgisayar ortamında da aktif olarak kullanılmaya başlanmıştır. 24 Bit Gerçek Renk derinliğine sahip görüntüler için RGB renk uzayı referans olarak kullanılmaktadır. 24 Bit'in 8 Biti kırmızı renk yoğunluğunu, 8 Biti yeşil renk yoğunluğunu, 8 Biti de mavi renk yoğunluğunu ifade etmek için kullanılmaktadır. Her bir renk yoğunluğunu ifade etmek için 8 Bit kullanıldığından, renk yoğunluğu değeri 0 ile 255 arası değerler alabilmektedir.

Çizelge 3.4. RGB renk uzayından örnek renkler ve R,G,B yoğunluk değerleri

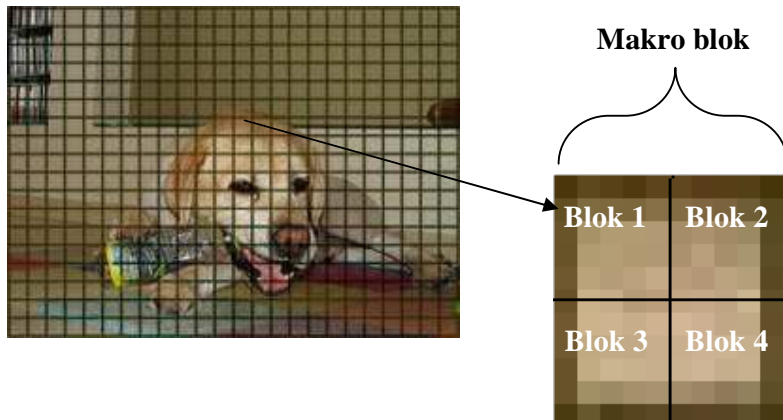
Renk	R yoğunluğu değeri	G yoğunluğu değeri	B yoğunluğu değeri
Kırmızı	255	0	0
Yeşil	0	255	0
Mavi	0	0	255
Siyah	0	0	0
Beyaz	255	255	255
Sarı	255	255	0
Camgöbeği	0	255	255
Eflatun	255	0	255



Çizelge 3.4'de RGB renk uzayından örnek renkler ve bu renklerin R, G, B yoğunluk değerleri görülmektedir. Bilgisayar ortamında farklı R, G, B yoğunluk değerleri verilerek 16.777.216 çeşit renk oluşturmak mümkündür.

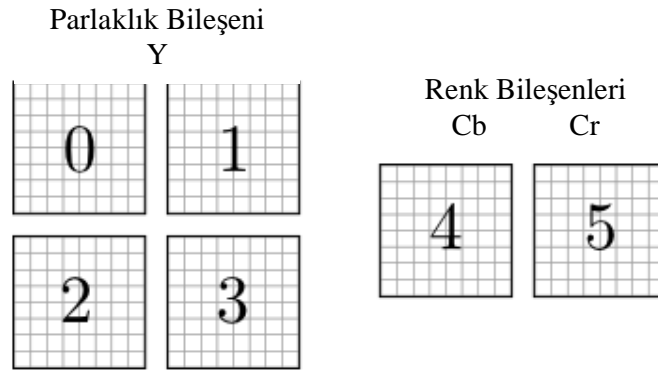
İnsan gözü, bir noktadaki parlaklık değişikliğini, renk değişikliğine göre daha çok fark etmektedir(Taşkın, 2007). Video görüntülerini kaydetmek ve görüntülemek için YUV renk uzayı kullanılmaktadır. YUV renk uzayında Y parlaklık değerini, U ve V ise renk bilgisini saklamaktadır. Dolayısıyla, YUV renk uzayı insan gözünün daha hassas olduğu parlaklık bilgisini referans almaktadır. MPEG-2 video sıkıştırma yönteminde, insan gözünün hassas olduğu bu nokta dikkate alınarak sıkıştırılmış dosya içersine piksellerin renk değerleri RGB renk uzayı yerine YUV renk uzayı kullanılarak kayıt edilmektedir. Bunun yanında renk bilgisi alt örnekleme işlemi yapılarak da sıkıştırılmış dosya içersine kayıt edilecek olan toplam renk bilgisi miktarının azaltılması sağlanmaktadır.

MPEG-2 video sıkıştırma yönteminde çerçeve içi kodlama yapılırken I-çerçevesi 8x8 piksel boyutunda bloklara ayrılmaktadır. Yan yana gelen 4 bloğa **makro blok** adı verilmektedir. Şekil 3.7'de bir resim çerçevesinin makro bloklara ayrılmış hali ve örnek bir makro bloğun da bloklara ayrılmış hali görülmektedir. Renk bilgisi alt örnekleme işlemi bu makro bloklar üzerinde yapılmaktadır.



Şekil 3.7. Blok ve makro bloklar

Makro blok 4 farklı bloktan oluşmaktadır. Renk bilgisi alt örnekleme işlemi yapılırken, makro bloğu oluşturan her bloğa ait parlaklık bilgisi (Y bileşeni) ayrı ayrı kayıt edilirken, her bloğa ait renk bilgisi (U ve V bileşenleri) yapılan örnekleme biçimine göre ayrı ayrı kayıt edilmemektedir. MPEG-2 sıkıştırma yönteminde kullanılan 4:2:0 renk bilgisi alt örnekleme biçiminde makro bloğu oluşturan her 4 blok için 4 farklı parlaklık bilgisi kayıt edilirken, sadece 2 farklı renk bilgisi 4 blok için ortak kullanılmak üzere kayıt edilmektedir.



Şekil 3.8. 4:2:0 Renk bilgisi alt örnekleme

Şekil 3.8’de 4:2:0 Renk bilgisi alt örnekleme işlemi görülmektedir. Bu örnekleme de makro bloğu oluşturan her bir blok için ayrı birer parlaklık bilgisi kayıt edilirken, mavi renk bilgisini tutan Cb renk bileşeni ile kırmızı renk bilgisini tutan Cr renk bileşenleri 4 blok için sadece 2 adet kayıt edilmektedir. Bu resim çerçevelerinin büyük oranda sıkıştırılmasını sağlamaktadır. Normal şartlarda 4 adet 8x8 piksel bloktan oluşan bir makro bloğun, RGB renk uzayında renk bilgisini depolamak için  $4 \times 8 \times 8 \times 3 = 768$  Bit’e ihtiyaç duyulurken, kabul edilebilir bir kayıp ile 4:2:0 renk bilgisi alt örnekleme yapılarak YUV renk uzayında depolamak için  $6 \times 8 \times 8 = 384$  Bit’e ihtiyaç duyulmaktadır. Bu %50 oranında sıkıştırma anlamına gelmektedir.

RGB renk uzayından YUV renk uzayına dönüşüm için kullanılan denklemler (3-1) aşağıda yer almaktadır.

$$\begin{aligned}
Y &= 0.229R + 0.587G + 0.114B \\
U &= -0.147R - 0.289G + 0.436B = 0.492(B-Y) \\
V &= 0.615R - 0.515G - 0.100B = 0.877(R-Y)
\end{aligned}
\tag{3-1}$$

Bu renk dönüşümü yapıldıktan sonra, renk bilgisi alt örnekleme biçimlerinden biri seçilerek sıkıştırma oranına büyük ölçüde artırılabilir. Renk bilgisi alt örnekleme biçimi olarak aşağıdaki biçimlerden biri kullanılmaktadır.

- 4:4:4
- 4:2:2
- 4:2:1
- 4:1:1
- 4:2:0
- 4:1:0

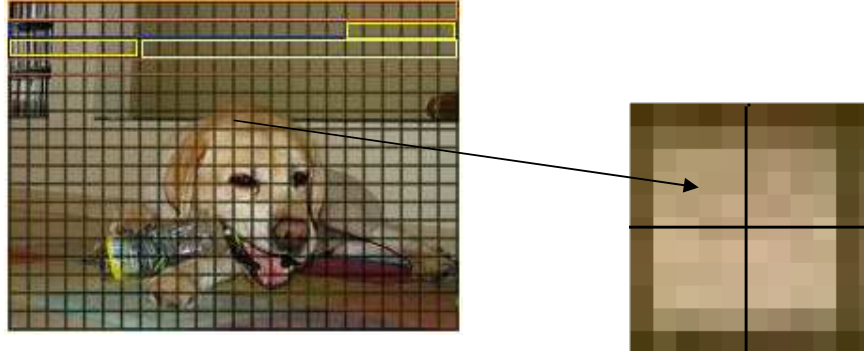
MPEG-2 video sıkıştırma yönteminde yer alan başlık bilgisinde, renk bilgisi alt örnekleme türlerinden hangisinin kullanıldığı belirtilmektedir.

### 3.3.1.2 Resim çerçevesinin 8x8 boyutunda bloklara ayrılması

Hareketli video görüntüleri oluşturan resim çerçeveleri 8x8 boyutunda bloklara ayrılmaktadır. 16x16 boyutunda olan makro bloklar iste 4 adet bloğun yan yana gelmesi ile oluşmaktadır. Çerçeve içi kodlama ve çerçeveler arası kodlama için de önemli ölçüde avantaj sağlayan bloklara ayırma işlemi, MPEG-2 video sıkıştırma yönteminin önemli noktalarından biridir. Çerçeveler arası kodlama yapılırken, resim çerçeveleri arasında sadece değişiklik gösteren makro bloklar kodlanarak önemli ölçüde sıkıştırma sağlanmaktadır.

Resim çerçeveleri için yapılan bu soyutlama işlemi, sadece bloklara ve makro bloklara ayırma seviyesinde değildir. Bloklara ayırma işlemi yapılırken, ilk olarak resim çerçevesi dilimlere bölünür. **Dilim** birden fazla makro bloğun birleşmesiyle oluşan resim çerçevesi parçası olarak tanımlanmaktadır. Şekil 3.9'da dilimlere ayrılmış örnek bir resim çerçevesi yer almaktadır. Dilimin başı ile sonu yatay olarak aynı konumda olmak zorunda değildir. Her bir dilim, 16x16 piksel boyutunda yer alan makro

bloklara, her bir makro blok da 8x8 piksel boyutunda 4 adet bloğa bölünmektedir. Bu soyutlama işlemi hareketli videonun kodlanması sırasında sıkıştırma yararına oldukça büyük avantaj sağlamaktadır.



Şekil 3.9. Resim çerçevesinin dilim, makro blok ve bloklara bölünmesi

### 3.3.1.3 Ayrık kosinüs dönüşümü

Resim çerçevelerinin makro bloklara ayrılmasından sonra, makro bloğu oluşturan her bir blok üzerinde, bloğu oluşturan piksellerin renk değerlerini frekans uzayında ifade edebilmek için 2 boyutlu ayrık kosinüs dönüşümü yapılmaktadır. Ayrık kosinüs dönüşümü, kayıpsız bir dönüşüm algoritmasıdır. 8x8 piksel boyutundaki bloklar üzerinde yapılan ayrık kosinüs dönüşümü ile yine 8x8 boyutunda bir dönüşüm katsayıları dizisi elde edilmektedir.

Aşağıdaki formülde (3-2) 8x8 pikselden oluşan blok üzerinde dönüşüm için kullanılan ayrık kosinüs dönüşümü formülü yer almaktadır.

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N}$$

$$u, v, x, y = 0, 1, 2, \dots, N-1$$

x, y örnek 8x8 blok üzerindeki koordinatları

u, v frekans uzayındaki koordinatları belirlemektedir (3-2)

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & u, v = 0 \\ 1 & u, v \neq 0 \end{cases}$$

8x8 blok üzerinde ayrık kosinüs dönüşümü yapılmadan önce, dönüşüm sonucunda elde edilecek dönüşüm katsayısı dizisi elemanlarının çok büyük değerler içermesine engel olmak ve birçok renk değerinin 0 civarında olmasını sağlamak amacıyla tüm renk değerlerinden 128 değeri çıkarılmaktadır. Tüm piksellerin renk değerinden 128 değeri çıkartıldığında yeni 8x8 matriste yer alacak olan piksellerin renk değeri [-128,127] arasında olacaktır.

$$\begin{array}{c}
 \begin{array}{cccccccc}
 & & & & x & & & \\
 & & & & \rightarrow & & & \\
 \begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix} & -128 = & \begin{array}{cccccccc}
 & & & & x & & & \\
 & & & & \rightarrow & & & \\
 \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} & \downarrow y & \\
 \end{array} \\
 \\
 \begin{array}{cccccccc}
 & & & & x & & & \\
 & & & & \rightarrow & & & \\
 \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix} & \downarrow y & \text{DCT} \rightarrow & \begin{array}{cccccccc}
 & & & & u & & & \\
 & & & & \rightarrow & & & \\
 \begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix} & \downarrow v & \\
 \end{array} \\
 \end{array}
 \end{array}$$

Şekil 3.10. Ayrık kosinüs dönüşümü

Şekil 3.10'da yer alan 8x8 blok üzerinde ayrık kosinüs dönüşümü uygulanmıştır. Dikkat edilirse, en yüksek mutlak değere sahip eleman sol üst köşede yer almaktadır. Dönüşüm sonunda elde edilen matriste yer alan bu elemana **DC katsayısı** adı verilmektedir. Kalan diğer 63 dizi elemanı ise **AC katsayısı** olarak adlandırılmaktadır.

### 3.3.1.4 Nicelendirme

Makro blokları oluşturan bloklar üzerinde ayrık kosinüs dönüşümü uygulandığında renk değerlerinin, frekans uzayındaki karşılıkları elde edilmektedir. Şekil 3.10'da yer alan sonuç matrise bakıldığında 415 değerinin yüksek bir değer olduğu ve bu değerın bilgisayar ortamında saklanabilmesi için 8 Bit'ten daha fazla Bit'e

ihtiyaç duyulduğu bilinmektedir. İnsan gözü, nispeten büyük bir alandaki ufak parlaklık değişikliklerini daha hassas olarak fark edebilirken, yüksek parlaklık çeşitliliğinin olduğu bir ortamda, parlaklığın gücünü kesin olarak tespit etmekte yetersiz kalmaktadır. Bu ayrıntı, yüksek frekanslı bileşenlerin daha az depolama alanı ile ifade edilmesine imkân sağlamaktadır.

Nicelendirme işlemi ile aslında, ayrık kosinüs dönüşümü yapılmış olan matris üzerinde basit bir bölme ve yuvarlama işlemi yapılmaktadır. Tüm matris değerleri, sabit bir sayıya bölünmekte ve sonuç en yakın tamsayıya yuvarlanmaktadır. Bu bölerek yuvarlama işlemi çerçeve içi kodlama esnasında veri kaybına neden olan tek işlemidir. Nicelendirme işlemi sonucunda, yüksek frekansa sahip bileşenler 0 değerine yuvarlanmaktadır ve yuvarlanan bu değerler, değişken uzunlukta kodlama safhasında sıkıştırma işleminde büyük bir avantaj sağlamaktadır.

Bölme işlemi sırasında kullanılan 8x8 boyutundaki matrise nicelendirme matrisi adı verilmektedir. MPEG-2 video sıkıştırma yönteminde, kullanıcı tarafından atanmış olan bir nicelendirme matrisi de kullanılabilir. Bu matrisin, sıkıştırılmış video içerisinde uygun bir başlıktan sonra yer alması gerekmektedir. Kullanıcı tarafından bir nicelendirme matrisinin sıkıştırılmış video dosyası içerisinde sunulmadığı durumlarda, çerçeve içi kodlama ve çerçeveler arası kodlama sırasında kullanılacak 2 farklı nicelendirme matrisi bulunmaktadır.

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

a. Çerçeve içi kodlamada kullanılan nicelendirme matrisi

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

b. Çerçeveler arası kodlamada kullanılan nicelendirme matrisi

Şekil 3.11. Standart nicelendirme matrisleri

Şekil 3.11’de standart olarak kullanılan nicelendirme matrisleri görülmektedir. Nicelendirme işleminin matematiksel formülü aşağıda gösterilmektedir.

$$B_{j,k} = \text{Yuvarla} \left( \frac{G_{j,k}}{Q_{j,k}} \right) \quad j=0,1,\dots,N-1 \quad k=0,1,2,\dots,N-1 \quad (3-3)$$

Bu denklemde (3-3)  $G_{j,k}$  ayrık kosinüs dönüşümü sonucunda elde edilmiş olan matris,  $Q_{j,k}$  nicelendirme matrisi ve  $B_{j,k}$  ise nicelendirme işlemi sonucunda elde edilen matristir.

$$Q_{j,k} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Şekil 3.12. Örnek nicelendirme matrisi

Ayrık kosinüs dönüşümü yapılmış olan matris dizisi üzerinde nicelendirme işlemi için, kullanıcı tarafından sıkıştırılmış video görüntüsü içerisinde yerleştirilmiş ve Şekil 3.12’de görülen örnek nicelendirme matrisi kullanıldığında, nicelendirme işlemi sonucunda Şekil 3.13’de yer alan sonuç matris elde edilmektedir.

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Şekil 3.13. Nicelendirme işlemi sonucunda elde edilen matris

Şekil 3.13’de yer alan nicelendirme işlem sonucunda elde edilen matriste birçok 0 değerinin yer aldığı görülmektedir. Bu işlemi sonucunda elde edilen bu matris üzerinde zigzag tarama işlemi uygulanarak, elde edilen tek boyutlu dizi üzerinde değişken uzunlukta kodlama yapılarak oldukça büyük bir sıkıştırma oranı yakalanmaktadır.

### 3.3.1.5 Zigzag tarama

Nicelendirilme işlemi sonucundan elde edilmiş olan sonuç matrisinin değişken uzunlukta kodlama safhasına geçmeden önce tek boyutlu bir dizi haline getirilmesi gerekmektedir. Zigzag tarama safhasında sonuç matris, tahmini olarak en düşük uzaysal frekansa sahip olandan en yüksek uzaysal frekansa sahip olana gidecek şekilde sıralanan tek boyutlu bir dizi haline getirilmektedir. Sıralama esnasında, kullanılan matrislere **tarama deseni** adı verilmektedir ve standart olarak zigzag tarama sırasında 2 farklı tarama deseni kullanılmaktadır. Şekil 3.14’de zigzag taramada kullanılan standart tarama desenleri görülmektedir.

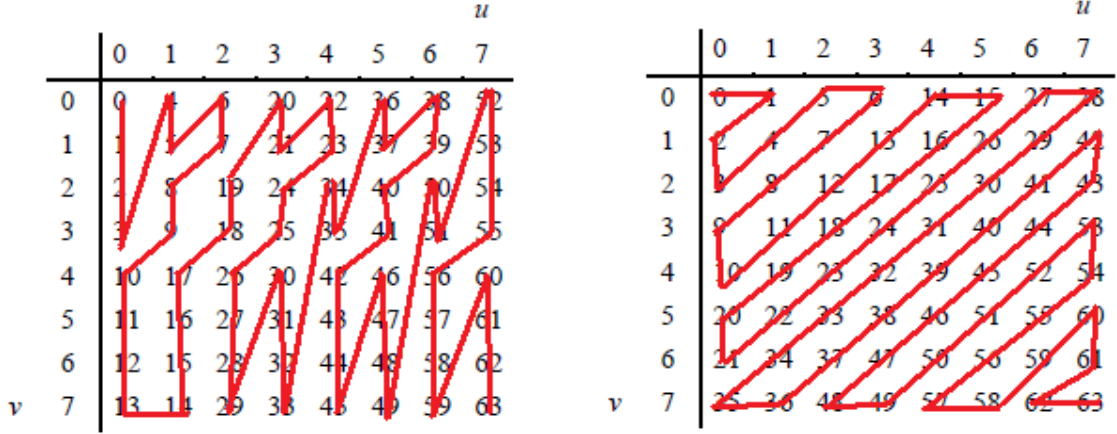
	<i>u</i>									<i>u</i>							
	0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7
0	0	1	5	6	14	15	27	28	0	0	4	6	20	22	36	38	52
1	2	4	7	13	16	26	29	42	1	1	5	7	21	23	37	39	53
2	3	8	12	17	25	30	41	43	2	2	8	19	24	34	40	50	54
3	9	11	18	24	31	40	44	53	3	3	9	18	25	35	41	51	55
4	10	19	23	32	39	45	52	54	4	10	17	26	30	42	46	56	60
5	20	22	33	38	46	51	55	60	5	11	16	27	31	43	47	57	61
6	21	34	37	47	50	56	59	61	6	12	15	28	32	44	48	58	62
<i>v</i> 7	35	36	48	49	57	58	62	63	<i>v</i> 7	13	14	29	33	45	49	59	63

Şekil 3.14. Zigzag taramada kullanılan tarama desenleri

Tarama desenleri, nicelendirme işlemi sonucundan elde edilmiş olan matrisin tek boyutlu dizi haline getirilmesi sırasında kullanılmaktadır.



Şekil 3.15’de zigzag tarama adımları görülmektedir. Görüldüğü üzere, nicelendirme matrisini tek boyutlu bir dizi haline getirirken, birbirini takip eden bir sıralama değil de tarama deseninde belirlenmiş olan sıra kullanılmaktadır.



Şekil 3.15. Zigzag tarama adımları

Şekil 3.13’ de yer alan nicelendirme işlemi sonucunda elde edilen sonuç matris zigzag tarama kullanılarak tek boyutlu bir dizi haline getiriliğinde aşağıdaki sonuç elde edilmektedir.

$$\begin{aligned} \text{QFS}[n] = & (-26, -3, 0, -3, -2, -6, 2, -4, 1, -4, 1, 1, 5, 1, 2, -1, 1, -1, 2, 0, 0, 0, 0, 0, \\ & -1, -1, 0, \\ & 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \end{aligned}$$

QFS[n] matrisinin 27. elemanından sonra yer alan tüm elemanlarının 0 değeri aldığı görülmektedir. MPEG-2 sıkıştırma yönteminde bu durum avantaj olarak kullanılmakta, değişken uzunlukta kodlama yapılırken, zigzag tarama sonucunda elde edilmiş olan tek boyutlu sonuç matrisinde belirli bir elemandan sonra gelen elemanlar 0 değerini alıyorsa, bu blok sonu işaretiyle belirlenip sıkıştırılmış dosya içerisinde kodlanmayarak sıkıştırma oranı arttırılmaktadır.

### 3.3.1.6 Değişken uzunlukta kodlama

Zigzag tarama aşamasının sonucunda elde edilen tek boyutlu matris üzerinde değişken uzunlukta kodlama işlemi yapılmaktadır. Değişken uzunlukta kodlanacak her bir kelimenin belirlenmesi için, kodlama noktasından sonra 0'dan farklı bir değere rastlayana kadar kaç adet 0 değeri olduğu sayılmaktadır.

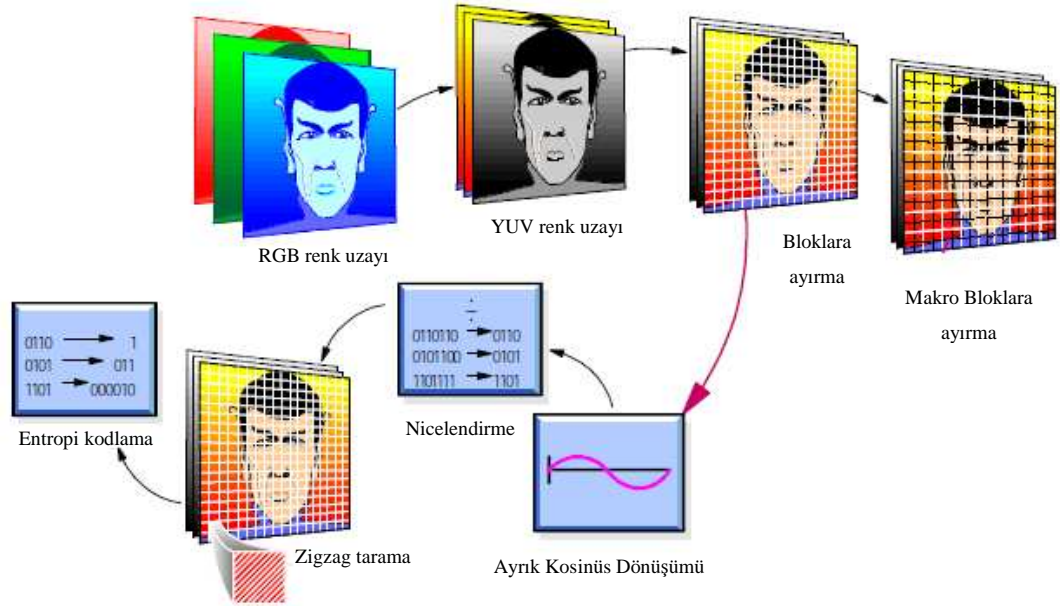
$$\begin{aligned} \text{QFS}[n] = & (-26, -3, 0, -3, -2, -6, 2, -4, 1, -4, 1, 1, 5, 1, 2, -1, 1, -1, 2, 0, 0, 0, 0, 0, \\ & -1, -1, 0, \\ & 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) \end{aligned}$$

Yukarıdaki matris dizisi değişken uzunlukta kodlamaya tabi tutulduğunda;

$$\begin{aligned} & (0,-26), (0,-3), (1,0), (0,-3), (0,-2), (0,-6), (0,2), (0,-4), (0,1), (0,1), (0,5), (0,1), \\ & (0,2), (0,-1), (0,1), (-0,1), (0,2), (5,-1), (0,-1), (0,-1), (\text{EOB}) \end{aligned}$$

kod kelimeleri elde edilmektedir. Bu kod kelimelerin Bit karşılıkları değişken uzunlukta kodlama tablosuna bakılarak belirlenmektedir(Ek A ve Ek B) ve her bir kod kelimesini ifade etmek için kullanılan Bit sayısı değişken uzunluktur.

Şekil 3.16'da çerçeve içi kodlamanın tüm aşamaları görülmektedir. Büyük sahne değişimlerinin söz konusu olduğu durumlarda, I-çerçeveleri oluşturulmaktadır ve bu aşamaların tamamı I-çerçeveleri üzerinde uygulanmaktadır.



Şekil 3.16. Çerçeve içi kodlamanın aşamaları

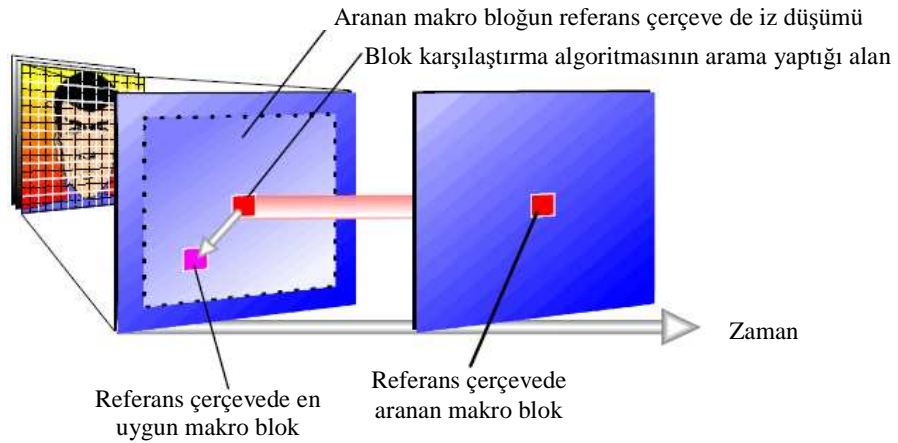
### 3.3.2. Çerçeveler arası kodlama

Çerçeveler arası kodlama işlemi B-çerçeveleri ve P-çerçeveleri üzerinde yapılmaktadır. Çerçeveler arası kodlama işleminde, çerçeve içi kodlama da olduğu gibi ilk olarak resim çerçevesi makro bloklara ve bloklara ayrılmaktadır. I-çerçevesini kodlarken ayırım yapılmaksızın tüm makro bloklar üzerinde çerçeve içi kodlama aşamaları yapılmaktadır. B-çerçevelerini ve P-çerçevelerini kodlarken ise, MPEG-2 sıkıştırma algoritması ilk olarak kodlanacak olan çerçeveden önce gelen çerçeveler içerisinde, kodlanacak çerçeve içerisinde yer alan makro blokların aynısının olup olmadığı kontrol etmektedir. Eğer kodlanacak çerçeveden önce gelen çerçeveler arasında, kodlanacak çerçevedeki makro bloklardan birinin aynısı yer alıyorsa, bu çerçeveye **referans çerçeve** adı verilmektedir. Referans çerçeveyi belirleme sırasında, çerçeveler arasında aynı renk değerlerine sahip makro blokların var olup olmadığının tespit edilmesi için **blok karşılaştırma algoritması** kullanılmaktadır. Blok karşılaştırma algoritmasının her zaman aynı renk değerlerine sahip blokları referans çerçeve ile kodlanan çerçeve içerisinde bulması mümkün olmamaktadır. Referans çerçeve ile kodlanan çerçeve arasında birine çok yakın renk değerleri içeren makro

blokların tespit edildiği durumlarda, makro bloklar arasındaki renk değerleri farkının da kodlanması gerekmektedir. **Tahmin hatası** olarak adlandırılan bu renk değerleri farklılığı kodlanmaktadır.

Blok karşılaştırma algoritması tarafında, kodlanacak çerçeve ile referans çerçeve içerisinde ortak bir makro blok bulunduğu, tespit edilen makro blok tekrar kodlanmamaktadır. Bu makro blok için sadece **hareket vektörü** olarak adlandırılan ve bu makro bloğun referans çerçevedeki konumuna işaret eden bir vektör kodlanmaktadır. Hareket vektörünün oluşturulması işlemine **hareket tahmini** adı verilmektedir.

Sonuç olarak, çerçeveler arası kodlama yapılırken, kodlanacak çerçeveler içerisindeki makro blokların daha önceki çerçeveler de yer alıp almadığı kontrol edilmektedir. Eğer kodlanacak çerçeve içerisinden yer alan makro bloklardan bir ile aynı renk değerlerine sahip bir çerçeve var ise bu çerçeve referans çerçeve olarak adlandırılmaktadır. Referans çerçeve içerisinde yer alan ortak makro blok ile kodlanacak çerçeve içerisinde yer alan makro blok arasında ufak bir farklılık söz konusu ise bu fark ile birlikte, makro bloğun referans çerçeve içerisinde yer alan konumunu gösteren hareket vektörü kodlanmaktadır.



Şekil 3.17. Çerçeveler arası kodlama

Şekil 3.17'de çerçeveler arası kodlama aşamaları görülmektedir. İlk olarak kodlanacak olan çerçeve içerisinde yer alan makro bloklar için, kendinden önce gelen

çerçeveler arasında yer alıp almadığına bakılmaktadır. Eğer kontrol edilen makro blok, kendinden önceki çerçeveler içersinde aynen yer almışsa, kodlanacak çerçeve için bu makro bloğun sadece referans çerçevedeki konumu kodlanmaktadır.

Hareketli video görüntülerinin saniye de ortalama 25 resim çerçevesi içerdiği düşünüldüğünde, çerçeveler arası kodlama sayesinde birçok resim çerçevesinde ortak olan makro blokların kodlamaması sağlanarak büyük ölçüde bilgi tekrarının önüne geçilmektedir.

P-çerçevesi referans çerçeve olarak, sadece kendinden önce gelen I-çerçevesini ya da P-çerçevesini kullanmaktadır. B-çerçevesi ise, referans çerçeve olarak kendinden önce ya da sonra gelen I-çerçevesini ya da P-çerçevesini kullanabilmektedir. Bu yüzden sıkıştırılmış dosya içersinde I, B ve P çerçevelerinin kaydedilme sırasıyla, gösterim sıraları farklılık gösterebilmektedir. Ayrıca, sadece çok büyük sahne değişimlerinin söz konusu olduğu durumlar da yaratılması düşünülen I-çerçevelerinin çok büyük sahne değişimleri olmasa dahi belirli bir periyotta yaratılması gerekmektedir. Çünkü P-çerçevesi referans çerçeve olarak sadece kendinden önce gelen I-çerçevelerini ya da P-çerçevelerini kullanmaktadır. Saniye de 25 resim çerçevesi içeren hareketli video görüntüsü için ortalama 12 saniyede bir I-çerçevesi yaratılmaktadır.

MPEG-2 sıkıştırma yönteminde, çerçeveler arası kodlamayı daha etkin bir biçimde yapabilmek için, resim çerçeveleri gruplandırılmaktadır. Çünkü referans çerçeve olarak, kodlaması yapılan çerçeveden önce gelen tüm çerçevelere bakmak sıkıştırma algoritmasının performansını büyük ölçüde azaltmaktadır. Çerçeveler arası kodlama sırasında blok karşılaştırma algoritması **resim grubu** olarak adlandırılan ve birden fazla sayıda I, B ve P-çerçevesinden oluşan bir resim dizisi üzerinde yapılmaktadır. Sıkıştırılmış dosya içersinde resim gruplarını birbirinden ayırt edebilmek için ise başlık bilgileri kullanılmaktadır.

#### 4. MPEG-2 VİDEO AKIMI

MPEG-2 yöntemi kullanılarak sıkıştırılan hareketli video görüntüsü **video akımı** olarak adlandırılan bir Bit dizisine dönüştürülmektedir. Sıkıştırılmış hareketli video görüntüsünün video akımına dönüştürülmesi, bir noktadan diğerine aktarımı sırasında, dosyanın tamamı aktarılmadan da görüntülenmesine imkân sağlamaktadır.

MPEG-2 yönteminin kullanmış olduğu çerçeve içi kodlama ile çerçeveler arası kodlamanın, dosya içersinde başladığı ve bittiği yerlerin belirtilmesi gerekmektedir. MPEG-2 yöntemi kullanarak sıkıştırma yapıldığında 3 farklı türde resim çerçevesi oluşturulmaktadır. Çerçeve içi kodlama I-çerçevesi üzerinde yapılmaktadır. I-çerçevesi dilimlere bölünmektedir. Çerçeve içi kodlama adımları her bir dilimi oluşturan makro bloklar üzerinde işletilmektedir. Her bir makro blok da bloklardan oluşmaktadır. Çerçeveler arası kodlama bir resim grubu içersinde yer alan I, B ve P-çerçeveleri kullanılarak yapılmaktadır. Video akımı içersinde bu resim grubunun hangi resim çerçevelerini içerdiğini belirtmesi gerekmektedir.

Video akımı sadece hareketli görüntünün oluşturulması için kullanılacak olan resim çerçevelerini değil aynı zamanda ses bilgisini de içermektedir. Akım içersinde yer alan ses bilgisinin de dosya içersindeki konumunun belirtilmektedir.

MPEG-2 video sıkıştırma yöntemi kullanılarak oluşturulan video akımı içersinde birden fazla yapı (resim grubu, resim çerçevesi, dilim, makro blok, blok, v.b.) yer almaktadır. Sıkıştırılmış dosya açılırken, çözücünün bu yapıların dosya içersindeki yerlerini belirleyerek, uygun çözüme algoritmalarını kullanması gerekmektedir. Çerçeve içi kodlama kullanılarak sıkıştırılmış olan bir I-çerçevesinin çözülmesi için kullanılacak olan algoritmanın adımları ile çerçeveler arası kodlama kullanılarak sıkıştırılmış olan bir B-çerçevesinin çözülmesi için kullanılacak algoritmanın adımlarının aynı olması mümkün değildir. MPEG-2 sıkıştırma yöntemi kullanılarak oluşturulmuş olan video akımı içersinde yer alan yapıları birbirinde ayırt edebilmek için dosya içersine kontrol kelimeleri eklenmektedir. Bu kontrol kelimelerine **başlangıç kodu** adı verilmektedir ve MPEG-2 sıkıştırma yönetimden her bir başlangıç kodu 4 Byte'tan oluşmaktadır.

Her bir başlangıç kodu, **başlangıç kodu ön eki** ve **başlangıç kodu değerinden** oluşmaktadır. Başlangıç kodu öneki 3 Byte'tır. Şekil 4.1'de başlangıç kodunun yapısı görülmektedir. Her bir başlangıç kodu için ön ek 23 adet 0 ve 1 adet 1 değeri içermektedir. Başlangıç kodu değeri ise 8 Bitten oluşmaktadır.

Başlangıç kodu ön eki			Başlangıç kodu değeri
0	0	1	x
0000 0000	0000 0000	0000 0001	xxxx xxxx

Şekil 4.1. Başlangıç kodu yapısı

MPEG-2 yönetiminde başlangıç kodunun aldığı değerler Çizelge 4.1'de gösterilmektedir.

Çizelge 4.1. MPEG-2 yönteminde kullanılan başlangıç kodları

Açıklama	Başlangıç kodu değeri (Onaltılık)	Başlangıç kodu değeri (Onluk)
Resim çerçevesi başlangıç kodu	00	0
Dilim başlangıç kodu	01'den AF'e kadar	1'den 175'e kadar
Ayrılmış kod	B0	176
Ayrılmış kod	B1	177
Kullanıcı verisi başlangıç kodu	B2	178
Video akımı başlangıç kodu	B3	179
Video akımı hatası başlangıç kodu	B4	180
Genişletme başlangıç kodu	B5	181
Ayrılmış kod	B6	182
Video akımı bitiş kodu	B7	183
Resim grubu başlangıç kodu	B8	184
Sistem başlangıç kodları	B9'dan FF'e kadar	185'den 255'e kadar

Çizelge 4.1’de gösterilen kodlar içerisinde yer alan **sistem başlangıç kodları**, birden fazla video akımının tek bir paket halinde bir yerden bir yere iletimi sırasında, video akımlarını birbirinden ayırt etmek için kullanılan kodlardır. MPEG-2 sıkıştırma yöntemi kullanılarak oluşturulan video akımı senkronize edilmiş ve paketlenmiş olarak birden fazla video ve ses akımını aynı anda içerebilmektedir. Dijital uydu alıcıları tarafından alınan sayısal bilgiler de bu paketlenmiş video akımından oluşmaktadır.

Tek bir hareketli video görüntüsü içeren MPEG-2 video akımı için hiyerarşik yapı aşağıda sıralı olarak belirtilmektedir.

- Video Akımı
- Video akımı başlığı
- Resim grubu başlığı
- Resim çerçevesi başlığı
- Dilim başlığı
- Makro blok başlığı
- Blok başlığı

Yukarıda belirtilmiş olan her bir hiyerarşik yapı elamanı çözümlenerek, sıkıştırılmış video görüntüsü ile ilgili detaylı bilgiler ve görüntünün kendisi elde edilmektedir. Bazen başlangıç kodlarının içerisinde tanımlanması mümkün olmayan bilgileri tanımlamak için genişletilmiş başlangıç kodlarına ihtiyaç duyulmaktadır.

#### **4.1. Video Akımı Başlığı**

MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içerisinde hem ses hem de video birbirini takip edecek biçimde karışık olarak bulunmaktadır. Sıkıştırılmış dosyanın çözümlenmesi sırasında, bu verileri birbirinden ayırt etmek için başlangıç kodları kullanılmaktadır.

Sıkıştırılmış video görüntüsünde, görüntü kodları en üst seviyede yer alan yapı video akımıdır. Dosya içerisinde, hareketli görüntü ile ilgili veriler video akımı başlangıç kodunun hemen ardında yer almaktadır. Şekil 4.2’de video akımı başlangıç kodu görülmektedir. Çözücü birbirini takip eden 24 Biti kontrol ederek, “0000 0000



0000 0000 0000 0001 1011 0011” deęeriyle karřılařırsa, dosya iersinde bu komundan sonra video akımı verisi yer alacaktır yorumunu yapmaktadır.

<b>Video akımı bařlangı kodu</b>			
0	0	1	179
0000 0000	0000 0000	0000 0001	1011 0011

řekil 4.2. Video akımı bařlangı kodu

Video akımı bařlangı kodu, özülecek olan video akımı hakkında bilgiler ieren ve video akımı bařlıęı olarak adlandırılan bir Bit dizisinin bařında yer almaktadır.

izelge 4.2’de video akımı bařlıęının detayları grlmektedir. Video akımı bařlıęı Bit dizisi iersinde, ereve ii kodlama ve ereveler arası kodlama iin standart nicelendirme matrislerinin dıřında bařka bir matris kullanılıp kullanılamayacaęı belirlenmektedir. Bu yzden video akımı Bit dizisinin uzunluęu deęiřkendir. Video akım, sıkıřtırılmıř video dosyası iersinde yeni bir bařlangı koduna rastlayana kadar devam etmektedir.

Çizelge 4.2. Video akımı başlığı

Başlık bilgisi	Bit sayısı
Video akımı başlangıç kodu	24 Bit
Yatay boyut değeri	12 Bit
Dikey boyut	12 Bit
En boy oranı bilgisi	4 Bit
Bir saniyede yenilenen çerçeve sayısı kodu	4 Bit
Bir saniyede işlenen Bit sayısı değeri	18 Bit
İşaret biti	1 Bit
Video önbelleği doğrulayıcı boyutu değeri	10 Bit
Sabit parametreler işareti	1 Bit
Çerçeve içi kodlama için nicelendirme matrisi var mı?	1 Bit
<i>Eğer (Çerçeve içi kodlama için nicelendirme matrisi var ise)</i>	
Çerçeve içi kodlama için nicelendirme matrisi [64]	8 x 64Bit
Çerçeveler arası kodlama için nicelendirme matrisi var mı?	1 Bit
<i>Eğer (Çerçeveler arası kodlama için nicelendirme matrisi var ise)</i>	
Çerçeveler arası kodlama için nicelendirme matrisi [64]	8 x 64Bit

Çizelge 4.2'ye bakıldığında çözülecek olan sıkıştırılmış video görüntüsü hakkında bilgi edinmek mümkün olmaktadır. Yatay boyut değeri ile dikey boyut değeri resim çerçevelerinin piksel türünden boyutunu ifade etmektedir. Bir saniyede işlenecek olan çerçeve sayısının değeri yerine kodu, video akımı başlığı içerisinde yer almaktadır. Çizelge 4.3'e bakılarak bir saniye de işlenecek olan çerçeve sayısı bulunmaktadır.

Çizelge 4.3. Bir saniyede işlenecek olan çerçeve sayısı tablosu

Saniye başına işlenecek çerçeve sayısı kodu	Saniye başına işlenecek çerçeve sayısı
0000	İzin verilmiyor
0001	24 000 / 1001 (23,976...)
0010	24
0011	25
0100	30 000 / 1001 (29,97...)
0101	30
0110	50
0111	60 000 / 1001 (59,94...)
1000	60
...	Ayrılmış değer
1111	Ayrılmış değer

Video akımı başlığına bakılarak, çerçeve içi kodlama ve çerçeveler arası kodlama için nicelendirme matrisi olarak standart matrislerin mi yoksa yine video başlığı içerisinde sunulmuş olan özel matrislerin mi kullanılacağına karar verilmektedir.

MPEG-2 sıkıştırma yöntemi kullanılarak oluşturulmuş olan dosyalar için, video akımı başlığından sonra yer alması gereken genişletilmiş video akımı başlangıç kodudur. Eğer dosya içerisinde video akımı başlangıç kodunun bitiminden sonra genişletilmiş video akımı başlangıç kodu gelmiyorsa, sıkıştırma yöntemi olarak MPEG-1 kullanılmış demektir.

#### 4.2. Genişletilmiş video akımı başlığı

MPEG-2 sıkıştırma yöntemi kullanılarak sıkıştırılmış dosyalar için video akımı başlığından hemen sonra genişletilmiş video akımı başlığı gelmektedir. MPEG-1 sıkıştırma yöntemi kullanılarak sıkıştırılmış dosyalar için ise video akımı başlığından hemen sonra genişletilmiş video akımı başlığının gelmesi zorunlu değildir.

Genişletilmiş video akımı başlığı 32 Bit uzunluğunda olan genişletilmiş video akımı başlık kodu ile başlamaktadır. Şekil 4.3’de genişletilmiş video akımı başlangıç kodu görülmektedir.

<b>Genişletilmiş video akımı başlangıç kodu</b>			
0	0	1	181
0000 0000	0000 0000	0000 0001	1010 0101

Şekil 4.3. Genişletilmiş video akımı başlangıç kodu

Çizelge 4.4’de genişletilmiş video akımı başlığının detayları yer almaktadır. Genişletilmiş video akımı başlığı içerisinde, video akımı başlığına sığmayan bilgiler bulunmaktadır. Özellikle renk bilgisi alt örnekleme formatı, genişletilmiş video akımı başlığına bakılarak belirlenmektedir.

Çizelge 4.4. Genişletilmiş video akımı başlığı

<b>Başlık bilgisi</b>	<b>Bit sayısı</b>
Genişletilmiş video akımı başlangıç kodu	32 Bit
Genişletilmiş başlangıç kodu belirleyicisi	4 Bit
Profil ve düzey belirleyicisi	8 Bit
Birbirini takip eden video akımı	1 Bit
Renk bilgisi alt örnekleme formatı	2 Bit
Dikey boyut genişletilme değeri	2 Bit
Yatay boyut genişletilme değeri	2 Bit
Birim başına Bit genişletilme değeri	12 Bit
İşaret biti	1 Bit
Düşük bekleme	8 Bit
Saniye başına çerçeve genişletme değeri_n	1 Bit
Saniye başına çerçeve genişletme değeri_d	2 Bit

Renk bilgisi alt örnekleme formatı, MPEG-2 video akımı çözülürken kullanılan önemli bir bilgidir. Çizelge 4.5’de renk bilgisi alt örnekleme formatının genişletilmiş video başlığından alabileceği değerler ve bu değerlerin karşılığında sıkıştırma sırasında kullanılmış olan renk bilgisi alt örnekleme formatı görülmektedir.

Çizelge 4.5. Renk bilgisi alt örnekleme formatı çizelgesi

00	Ayrılmış
01	4:2:0
10	4:2:2
11	4:4:4

MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içerisinde genişletilmiş video akımı başlığı okunduktan sonra, resim gruplarının ve resim grupları içerisinde yer alan resim çerçevelerinin çözülmesi işlemine geçilmektedir.

#### 4.3. Resim Grubu başlığı

Resim grubu başlığı kullanılarak, çerçeve içi ve çerçeveler arası kodlama yapılarak sıkıştırılmış olan resim çerçevelerinin dosya içerisindeki konumu belirlenmektedir. Özellikle çerçeveler arası kodlamanın verimliliğini arttırmak için resim çerçeveleri gruplanarak sıkıştırılmış dosya içersine kayıt edilmektedir.

Resim grubu başlığı, 32 Bitten oluşan resim grubu başlık kodu ile başlamaktadır. Şekil 4.4’de resim grubu başlangıç kodu görülmektedir. Çizelge 4.6’da resim grubu başlığının detayları yer almaktadır.

Resim grubu başlangıç kodu			
0	0	1	181
0000 0000	0000 0000	0000 0001	1010 0101

Şekil 4.4. Resim grubu başlangıç kodu

Çizelge 4.6. Resim grubu başlığı

Başlık bilgisi	Bit sayısı
Resim grubu başlangıç kodu	32 Bit
Zaman kodu	25 Bit
Kapalı resim grubu işareti	1 Bit
Kırılmış bağlantı işareti	1 Bit

Resim grubu başlığında, resim grubu başlangıç kodunun hemen ardından 25 Bit ile ifade edilen **zaman kodu** yer almaktadır. Zaman kodu;

- Çerçeve düşürme bayrağı
- Saat zaman kodu
- Dakika zaman kodu
- İşaret biti
- Saniye zaman kodu
- Resim zaman kodu

alanlarından oluşmaktadır. Çizelge 4.7’de zaman kodunu oluşturan 25 Bitin detayları görülmektedir. Resim grubu başlığı içersinde yer alan zaman koduna bakılarak, işlenecek olan resim grubunun hareketli video görüntüsünün hangi saat, dakika ve saniyesinde gösterilmesi gerektiğine karar verilmektedir.

Çizelge 4.7. Zaman kodu

Zaman kodu	Değer aralığı	Bit sayısı
Çerçeve düşürme bayrağı		1 Bit
Saat zaman kodu	0-23	5 Bit
Dakika zaman kodu	0-59	6 Bit
İşaret biti		1 Bit
Saniye zaman kodu	0-59	6 Bit
Resim zaman kodu	0-59	6 Bit

Çerçeve düşürme bayrağı sadece saniye başına işlenecek olan çerçeve sayısı 29.97 değerini aldığı durumlarda 1 değerini almaktadır.

Resim başlığında yer alan **kapalı resim grubu işareti**, resim grubu içerisinde yer alan tüm resimlerin, bu resim grubundan önceki resim grubundan bağımsız olarak çözülebildiği durumlarda 1 değerini almaktadır. Eğer resim grubu içerisinde yer alan herhangi bir resim çerçevesini çözebilmek için, bir önceki resim grubunda yer alan resim çerçevelerinden herhangi birine dahi ihtiyaç duyulursa, kapalı resim grubu işareti 0 değerini almaktadır.

Kırılmış bağlantı işareti, resim grubu içerisinde I-çerçevesinden sonra gelen ilk B-çerçevesinin çözülebilmesi için gerekli olan farklı türdeki bir çerçevenin hatalı çözülmelerinden ötürü, bu B-çerçevesinin çözülemeyeceği durumlarda 1 değerini almaktadır.

#### 4.4. Resim Başlığı

MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içerisinde resim grubunu oluşturan her bir resim çerçevesi için ayrı bir resim başlığı yer almaktadır. Resim başlığı 32 Bitten oluşan **resim başlangıç kodu** ile başlamaktadır. Şekil 4.5’de resim başlığı görülmektedir.

Resim başlığı			
0	0	1	0
0000 0000	0000 0000	0000 0001	0000 0000

Şekil 4.5. Resim başlığı

Çizelge 4.8’de resim başlığı görülmektedir. Resim başlığı içerisinde, resim çerçevesi için atanmış olan geçici referans numarası, resim çerçevesi tipi, video tampon doğrulayıcısı gecikmesi ve MPEG-1 sıkıştırma yönteminde kullanılan bazı özel bilgiler yer almaktadır.

Çizelge 4.8. Resim başlığı

<b>Başlık bilgisi</b>	<b>Bit sayısı</b>
Resim başlangıç kodu	32 Bit
Geçici referans numarası	10 Bit
Çerçeve kodlama tipi	3 Bit
Video tampon doğrulayıcısı gecikmesi	16 Bit
<i>Eğer (Çerçeve kodlama tipi 2 ya da 3 ise)</i>	
Tam_görüntüögesi_ileri_vektör	1 Bit
İleri_f_kodu	3 Bit
<i>Eğer (Çerçeve kodlama tipi 3 ise)</i>	
Tam_görüntüögesi_geri_vektörü	1 Bit
İleri_f_kodu	3 Bit
Resim ekstra Biti	1 Bit
<i>Eğer (resim ekstra Biti 1 ise)</i>	
Ekstra resim bilgisi	8 Bit

Geçici referans numarası MPEG-2 kodlayıcı tarafından her bir resim çerçevesine otomatik olarak atanan bir değerdir ve resim başlığı içerisinde 10 Bit ile ifade edilmektedir. Geçici referans numarası ilişkisel veritabanı modelinde, bir tabloda yer alan kayıtları birbirinden ayırt etmek için kullanılan anahtar değere benzetilebilir. Resim başlığı içerisinde yer alan çerçeve kodlama tipi, resim çerçevesinin tipini belirlemek için kullanılır ve 3 Bit'ten oluşmaktadır. Çizelge 4.9'da yer alan çerçeve tipi kodlama tablosuna bakılarak, resim çerçevesinin tipini belirlemek mümkün olmaktadır. MPEG-2 sıkıştırma yönteminde I-çerçevesi, B-çerçevesi ve P-çerçevesi olmak üzere 3 farklı türde resim çerçevesi bulunmaktadır. Dolayısıyla, kodlanacak olan resim çerçevesinin bu 3 türden biri ile eşleşmesi zorunludur.



Çizelge 4.9. Çerçeve tipi kodlama tablosu

Çerçeve tipi kodu	Çerçeve tipi
000	İzin verilmez
001	I-çerçevesi
010	P-çerçevesi
011	B-çerçevesi
100	D-çerçevesi (MPEG-1 de kullanılır)
101	Ayrılmış değer
110	Ayrılmış değer
111	Ayrılmış değer

I-çerçevelerinin tamamı sıkıştırılmış dosya içerisinde kodlanırken, B-çerçevesi ile P-çerçevesinin tamamı kodlanmamaktadır. Bu tez çalışmasında, MPEG-2 yöntemi kullanılarak sıkıştırılmış hareketli video görüntüsünü oluşturan resim çerçevelerinin içerisinde yer alan metin ifadeler tespit edilerek, içerik tabanlı sorgulama yapabilen bir Video OCR Sistemi gerçekleştirilmektedir. Bu sistem, MPEG-2 yöntemi kullanılarak sıkıştırılmış hareketli video görüntüsü içeren dosya içersine yer alan I-çerçevelerini, dosyanın tamamını açmadan elde ederek, metin tanıma işlemini bu resim çerçeveleri üzerinde yapmaktadır. Bu yüzden, MPEG-2 sıkıştırma yönteminin sadece I-çerçevelerini kodlayan ve çözümleyen bölümleri ele alınmaktadır.

#### 4.5. Genişletilmiş Resim Başlığı

MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içerisinde, resim başlığından sonra **genişletilmiş resim başlığı** yer alabilir. Resimler kodlanırken birden fazla parametre kullanılmaktadır. Bu parametreler arasında, resimler çözümlenirken de kullanılan nicelendirme kodu türü, çerçeve içi değişken uzunlukta kodlama biçimi, renk bilgisi alt örnekleme formatı parametreleri sayılabilir. Resimler kodlanırken kullanılmış olan bu ve benzeri parametreler genişletilmiş resim başlığı içerisinde yer almaktadır.

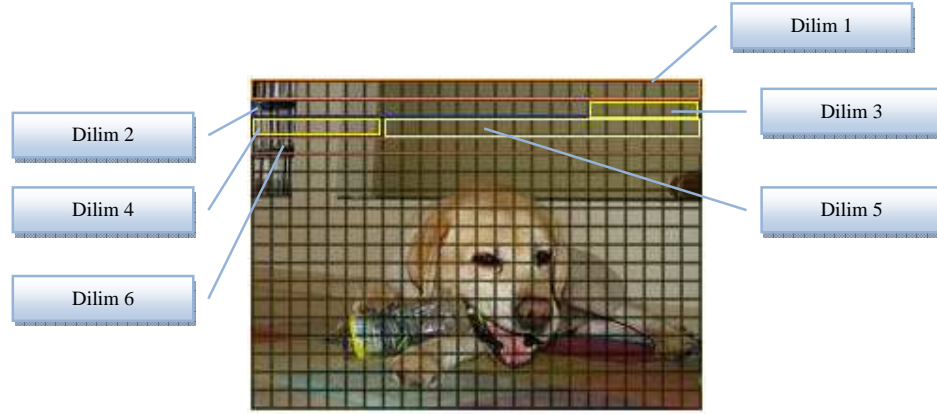
Çizelge 4.10. Genişletilmiş resim başlığı

Başlık bilgisi	Bit sayısı
Genişletilmiş başlangıç kodu	32 Bit
Genişletilmiş başlangıç kodu belirleyicisi	8 Bit
İleri yönde yatay hareket tahmini	4 Bit
İleri yönde dikey hareket tahmini	4 Bit
Geri yönde yatay hareket tahmini	4 Bit
Geri yönde dikey hareket tahmini	4 Bit
Çerçeve içi DC katsayısı duyarlılık değeri	2 Bit
Resim çerçevesi yapısı	2 Bit
İlk üst saha işareti	1 Bit
Çerçeve tahmini için ayrı kosinüs değeri	1 Bit
Gizleme hareket vektörü işareti	1 Bit
Nicelendirme ölçütü türü	1 Bit
Çerçeve içi değişken uzunlukta kodlama biçimi	1 Bit
Alternatif tarama yöntemi işareti	1 Bit
İlk sahayı tekrarla işareti	1 Bit
4:2:0 renk bilgisi alt örnekleme formatı işareti	1 Bit
İlerleyen çerçeve işareti	1 Bit
Karşıt görüntü işareti	1 Bit
<i>Eğer (karşıt görüntü işareti değeri 1 ise)</i>	
Dikey eksen	
Saha sırası	
Patlama şiddeti	
Alt taşıma safhası	

Çizelge 4.10'da genişletilmiş resim başlığı görülmektedir. Genişletilmiş resim başlığında yer alan, nicelendirme ölçütü türü, çerçeve içi değişken uzunlukta kodlama biçimi, alternatif tarama yöntemi işareti ile 4:2:0 renk bilgisi alt örnekleme formatı işareti I-çerçevelerinin çözümlenmesi sırasında önem teşkil eden parametrelerdir.

#### 4.6. Dilim Başlığı

MPEG-2 sıkıştırma yönteminde çerçeve içi ve çerçeveler arası kodlama yapılırken, resim çerçevesi dilimlere bölünmektedir. Her bir dilim de 16x16 piksel boyutunda makro bloklardan oluşmaktadır. Her bir makro blok da 8x8 piksel boyutunda 4 bloktan oluşmaktadır. Şekil 4.6'da dilimlere bölünmüş örnek resim çerçevesi görülmektedir. Her bir dilim makro bloklardan oluştuğu için yükseklik değeri makro bloğun yükseklik değeri olan 16 piksel olmak zorundadır.



Şekil 4.6. Dilimlere bölünmüş örnek resim çerçevesi

Her bir dilim başlığı, 32 Bitten oluşan dilim başlangıç kodu ile başlamaktadır. Şekil 4.7'de dilim başlangıç kodu görülmektedir. Dilimin resim çerçevesi içerisindeki yatay konumu, dilim başlangıç koduna bakılarak belirlenmektedir. Dilim başlangıç kodunun en sonunda yer alan 8 Bit, dilimin resim çerçevesi içerisindeki yatay konumunu belirlemek için kullanılmaktadır.

Dilim başlangıç kodu			
0	0	1	1 - 175
0000 0000	0000 0000	0000 0001	xxxx xxxx

Şekil 4.7. Dilim başlangıç kodu

Resim çerçevesini kodlamak için 175'ten daha fazla dilime ihtiyaç duyulduğunda, dilim başlığında yer alan genişletilmiş dilim yatay konum bilgisini içeren Bitlere bakılmaktadır. Çizelge 4.11'de dilim başlığı görülmektedir. Dilim başlığına bakılarak, dilimin çerçeve içi kodlama ya da çerçeveler arası kodlama yapılarak sıkıştırılmış makro bloklar içerip içermediğine karar verilebilmektedir. Ayrıca dilim başlığı içerisinde yer alan nicelendirme ölçek kodu, dilimi oluşturan makro blok ve bloklar çözülürken bir nicelendirme ölçek kodu bilgisine yer verilmedi ise kullanılmaktadır. Dilim çözme işlemi, yeni bir başlangıç kodu ile karşılaşılan kadar devam etmektedir.

Çizelge 4.11. Dilim başlığı

Başlık Bilgisi	Bit sayısı
Dilim başlangıç kodu	32 Bit
<i>Eğer (resim çerçevesinin yüksekliği &gt; 2800 ise) (Her bir makro blok 16 piksel boyutundadır. Resim çerçevesini kodlamak için 175 'den fazla dilim gerekiyorsa, resim boyutu 16*175 = 2800' den büyüktür.)</i>	
Genişletilmiş dilim yatay konumu bilgisi	3 Bit
Nicelendirme ölçek kodu	5 Bit
<i>Eğer (bir sonraki Bit 1 ise) {</i>	
Çerçeve içi kodlanacak dilim işareti	1 Bit
Çerçeve içi kodlanacak dilim	1 Bit
Ayrılmış bitler	7 Bit
<i>(bir sonraki Bit değeri 1 olduğu sürece){</i>	
Ekstra dilim bilgisi işareti	1 Bit
Ekstra dilim bilgisi	8 Bit
}	
}	
Ekstra dilim biti (0 değerinde)	1 Bit
Bir sonraki Bit dizisi 000 0000 0000 0000 0000 0000 olana kadar makro blok kodlama yap	

#### 4.7. Makro Blok Kodlama

Resim çerçevesini oluşturan dilimler 16x16 piksel boyutunda makro bloklardan oluşmaktadır. Her bir resim grubunun, resim çerçevesinin, resim çerçevesini oluşturan dilimlerini kodlamak için gerekli olan bilgilerin hangi konumdan başlayarak yer aldığı başlık bilgilerine bakılarak bulunabilmektedir. Buna karşın makro blokları ve blokları kodlarken başlık bilgileri kullanılmamaktadır. Dosya içerisinde bir dilim başlığı ile karşılaşıldığında, bir sonraki başlangıç koduna rastlanıncaya kadar, bu dilimi kodlamak için gerekli olan bilgilerin yer aldığı kabul edilmektedir. Özellikle, makro blokları oluşturan bloklar kodlanırken kullanılan değişken uzunlukta kodlamadan dolayı, dilimleri kodlarken kullanılacak olan Bit sayısı çok büyük farklılık göstermektedir.

Çizelge 4.12. Makro blok kodlama

Açıklama	Bit Sayısı
<i>Bir sonraki Bit dizisi 0000 0001 000 olduğu sürece {</i>	
Makro blok kaçış kodu	11 Bit
Makro blok adres artırımı	1-11 Bit
Makro blok tipi belirle ()	
<i>Eğer (makro blok tipi nicelendirme ölçütü içeriyorsa)</i>	
Nicelendirme ölçek kodu	5 Bit
<i>Eğer (makro blok tipi ileri yönde hareket bilgisi içeriyorsa)</i>	
Hareket vektörü (0)	
<i>Eğer (makro blok tipi geri yönde hareket bilgisi içeriyorsa)</i>	
Hareket vektörü (1)	
<i>Eğer (makro blok çerçeve içi kodlamada kullanılıyor ve gizli hareket vektörleri içeriyorsa)</i>	
İşaret Biti	1 Bit
<i>Eğer (makro blok desen içeriyorsa)</i>	
Makro blok deseni kodla()	
For (int i=0;i<blok sayısı;i++)	
Blok kodlama (i)	

Çizelge 4.12’de makro blok kodlama aşamaları görülmektedir. Makro blok kodlama işlemi, dosya içerisinde yeni bir başlangıç kodu ile karşılaşınca kadar sürmektedir. **Makro blok kaçış kodu**, kodlanacak makro blok ile bu makro bloktan önce kodlanan makro blok arasındaki makro blok adres değerinin 33’ten fazla olduğu durumlarda sıkıştırılmış dosya içerisinde yer alan 11 Bitten oluşan bir Bit dizisidir. Makro blok kaçış kodu “0000 0001 000” değerine sahiptir. Makro blok kodlama içerisinde yer alan **makro blok adres artırımı** ise kodlaması yapılan makro blok ile bu makro bloktan önce kodlanmış olan makro blok adres değeri arasındaki farktır. **Makro blok adresi**, kodlaması yapılan makro bloğun, resim çerçevesi içerisindeki konumunu ifade eden bir değişkendir. Resim çerçevesinin sol üst köşesinde yer alan makro bloğun adres değeri 0 olarak kabul edilmektedir.

Makro blok adres artım değerini kodlamak için kullanılacak Bit sayısı sabit değildir. Çizelge 4.13’de makro blok adres artım değeri için kullanılacak veriler bulunmaktadır. Çoğu zaman makro blok adres artım değeri 1 değerini almaktadır. Bu yüzden, makro blok adres artım değeri sıkıştırılmış dosya içersine değişken uzunlukta kodlama sistemi kullanılarak kaydedilmektedir. Makro blok adres artım değeri 1 ile 33 arası değerler almaktadır. Bu adres artım değerini sabit uzunlukta kodlama kullanarak ifade etmek için 6 Bit gerekmektedir. Her seferinde makro blok adres artım değerini ifade etmek için 6 Bit kullanmak yerine, değişken uzunlukta kodlama yapılarak, makro blok adres artım değerini kodlamak için farklı sayıda Bit kullanılmaktadır.

Çizelge 4.13. Makro blok adres artım değeri için değişken uzunlukta kodlama tablosu

Makro blok adres artım değeri için değişken uzunlukta kod	Artım değeri	Makro blok adres artım değeri için değişken uzunlukta kod	Artım değeri
1	1	0000 0101 01	18
011	2	0000 0101 00	19
010	3	0000 0100 11	20
0011	4	0000 0100 10	21
0010	5	0000 0100 011	22
0001 1	6	0000 0100 010	23
0001 0	7	0000 0100 001	24
0000 111	8	0000 0100 000	25
0000 110	9	0000 0011 111	26
0000 1011	10	0000 0011 110	27
0000 1010	11	0000 0011 101	28
0000 1001	12	0000 0011 100	29
0000 1000	13	0000 0011 011	30
0000 0111	14	0000 0011 010	31
0000 0110	15	0000 0011 001	32
0000 0101 11	16	0000 0011 000	33
0000 0101 10	17	0000 0001 000	Makro blok kaçış kodu

Resim çerçeveleri türleri arasında, en çok makro blok tanımlamanın yapıldığı çerçeve türü I-çerçevesidir. Aynı zamanda I-çerçevesi kodlanırken makro blok adres artım değeri sürekli 1 değerini almaktadır. Dolayısıyla, değişken uzunlukta kodlama kullanılarak, çoğu zaman 1 değerini almış olan makro blok adres artım değerini ifade edebilmek için sadece 1 Bite ihtiyaç duyulmaktadır.

#### 4.7.1. Makro blok tipi belirle() alt yordamı

Makro bloklar kodlanırken, **makro blok tipi belirle()** alt yordamı kullanılarak kodlaması yapılacak olan makro bloğun tipi belirlenmektedir. Makro blok tipleri, I-

çerçevesi, B-çerçevesi ve P-çerçevesi için farklılık göstermektedir. Bu tez çalışmasında sadece I-çerçevesi içerisinde kullanılabilen olan makro blok tipleri hakkında bilgi verilmektedir. I-çerçeveleri kodlanırken 2 farklı makro blok tipi kullanılmaktadır. Bu makro blok tipleri aşağıda listelenmiştir.

- Çerçeve içi kodlanmış makro blok
- Nicelendirme ölçütlü çerçeve içi kodlanmış makro blok

**Çerçeve içi kodlanmış makro blok** için Çizelge 4.12’de anlatılan makro blok kodlama aşamalarından sadece blok kodlama işlemi yapılmaktadır. **Nicelendirme ölçütlü çerçeve içi kodlanmış makro blok** için ise blok kodlama işleminin yanında, 5 Bitlik nicelendirme ölçütü de kodlama yapılmadan önce belirlenmektedir. Eğer makro blok kodlama sırasında, nicelendirme ölçütü yer almıyorsa, dilim başlığında belirlenmiş olan nicelendirme ölçütü kullanılmaktadır. I-çerçevesi için makro blok tipi belirleme işlemi 1 ile 2 Bit arasında değişken uzunlukta kodlanmış bir Bit dizisi ile ifade edilmektedir. Eğer bu değer “1” ise kodlanacak olan makro blok çerçeve içi kodlanmış makro blok, “01” ise nicelendirme ölçütlü çerçeve içi kodlanmış makro blok tipindedir.

Makro blok kodlama işleminin alt yordamı olan **blok kodlama()** yordamı, MPEG-2 video sıkıştırma yönteminde tanımlanmış olan en alt düzeydeki yapı olan blokları kodlamak için kullanılmaktadır. Bu kodlama işlemi, Çizelge 4.12’de görüldüğü gibi bir döngü kurularak makro bloğun içerdiği olduğu her bir blok için ayrı ayrı yapılmaktadır. Bir makro bloğun kaç adet blok içerdiği ise, resim bilgisi alt örnekleme tipine bağlı olarak farklılık göstermektedir. Video sıkıştırma işlemi yapılırken, renk bilgisi alt örnekleme formatı olarak 4:2:0 (4 Y, 1 Cb, 1 Cr) kullanıldığında, her bir makro blok 6 bloktan oluşmaktadır. Bu 6 bloktan, ilk 4’ü parlaklık bilgisini içeren bloklar iken, kalan 2’si ise renk bilgisini içeren bloklardır. **Blok kodlama()** yordamı, işlemler olduğu bloğun tipine göre (parlaklık bilgisi içeren blok ya da renk bilgisi içeren blok) farklı işlemler yaparak çözümlenme işlemini gerçekleştirmektedir.



#### 4.8. Blok kodlama() Yordamı

Makro blokları oluşturan her bir blok kodlanırken, Entropi kodlama kullanarak değişken uzunlukta veri dizileri elde edilerek sıkıştırılmış dosya içersine kayıt edilmektedir. Kodlanacak blok sayısı, kullanılmış olan renk bilgisi alt örnekleme biçimine göre farklılık göstermektedir.

Şekil 4.8’de parlaklık bilgisi içeren blokların kodlanma sırası, blokların DC ve AC katsayıları görülmektedir. Sıkıştırılmış dosya içersinde DC katsayılarının ve AC katsayılarının kodlanma türü farklıdır.

DC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC

DC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC

DC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC

DC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC
AC	AC	AC	AC	AC	AC	AC	AC

Şekil 4.8. Parlaklık bilgisi içeren bloklar

Çizelge 4.14’de ise **Blok kodlama()** yordamı yer almaktadır. **Blok kodlama()** yordamı parametre olarak bir tamsayı değişken almaktadır. Bu değişken 4 değerinden küçük ise yordam parlaklık bilgisi içeren bir bloğu çözümlenmektedir. Şayet 4 ve 4’ten büyük bir değer yordama parametre olarak yollanırsa, yordam renk bilgisi içeren bir bloğu çözümlenmektedir. Ayrıca kodlama yapılırken, her bloğa ait DC katsayısı değeri kodlanmayarak, DC katsayısı değerinin bir önce kodlanmış olan bloğun DC katsayısı

değeri ile farkı kodlanmaktadır. Aynı DC katsayısı değerine sahip ve ardı sıra kodlanmış bloklar için DC katsayı farkı boyutu 0 olarak kodlanarak, DC katsayısının kodlanmaması sağlanmaktadır. Bu benzer değerler tekrar tekrar kodlanmayarak, ya da sadece katsayılar arasındaki farklılık kodlanarak sıkıştırma oranını artırılmaktadır.

Çizelge 4.14. Blok kodlama() yordamı

Blok kodlama (i){	
<i>Eğer (makro blok tipi çerçeve içi kodlanmış makro blok ise)</i>	
Eğer (i<4){	
Parlaklık bilgisi içeren blok için DC katsayısı değeri boyutu	2-9 Bit
<i>Eğer (Parlaklık bilgisi içeren blok için DC katsayısı değeri boyutu 0 değil ise)</i>	
Parlaklık bilgisi içeren blok için DC katsayısı farkı	1-11 Bit
}	
Değilse {	
Renk bilgisi içeren blok için DC katsayısı değeri boyutu	2-10 Bit
<i>Eğer (Renk bilgisi içeren blok için DC katsayısı değeri boyutu 0 değil ise)</i>	
Renk bilgisi içeren blok için DC katsayısı farkı	1-11 Bit
Blok sonu kodu okunana kadar{	
Sonradan gelen katsayı değerleri	
}	

Çizelge 4.14'de yer alan parlaklık bilgisi içeren blok için DC katsayısı değer farkını elde ederken kullanılacak Bit sayısı Çizelge 4.15'de yer alan tabloya bakılarak tespit edilmektedir. Aynı şekil de Çizelge 4.14'de yer alan renk bilgisi içeren blok için DC katsayı değer farkını elde ederken kullanılacak Bit sayısı Çizelge 4.16'de yer alan tabloya bakılarak tespit edilmektedir.

Blokların AC katsayılarını kodlamak için Huffman kodlamanın bir çeşidi olan **seri düzey kodlama** kullanılmaktadır. Bu kodlama türü kullanılarak elde edilen veriler değişken uzunlukta kodlama kullanılarak sıkıştırılmış dosya içersine kaydedilmektedir.

Çizelge 4.15. Parlaklık bilgisi içeren bloklar kodlanırken DC katsayısı farkını kodlamakta kullanılacak Bit sayısı

100	0
00	1
01	2
101	3
110	4
1110	5
1111 0	6
1111 10	7
1111 110	8
1111 1110	9
1111 1111 0	10
1111 1111 1	11

Çizelge 4.16. Renk bilgisi içeren bloklar kodlanırken DC katsayısı farkını kodlamakta kullanılacak Bit sayısı

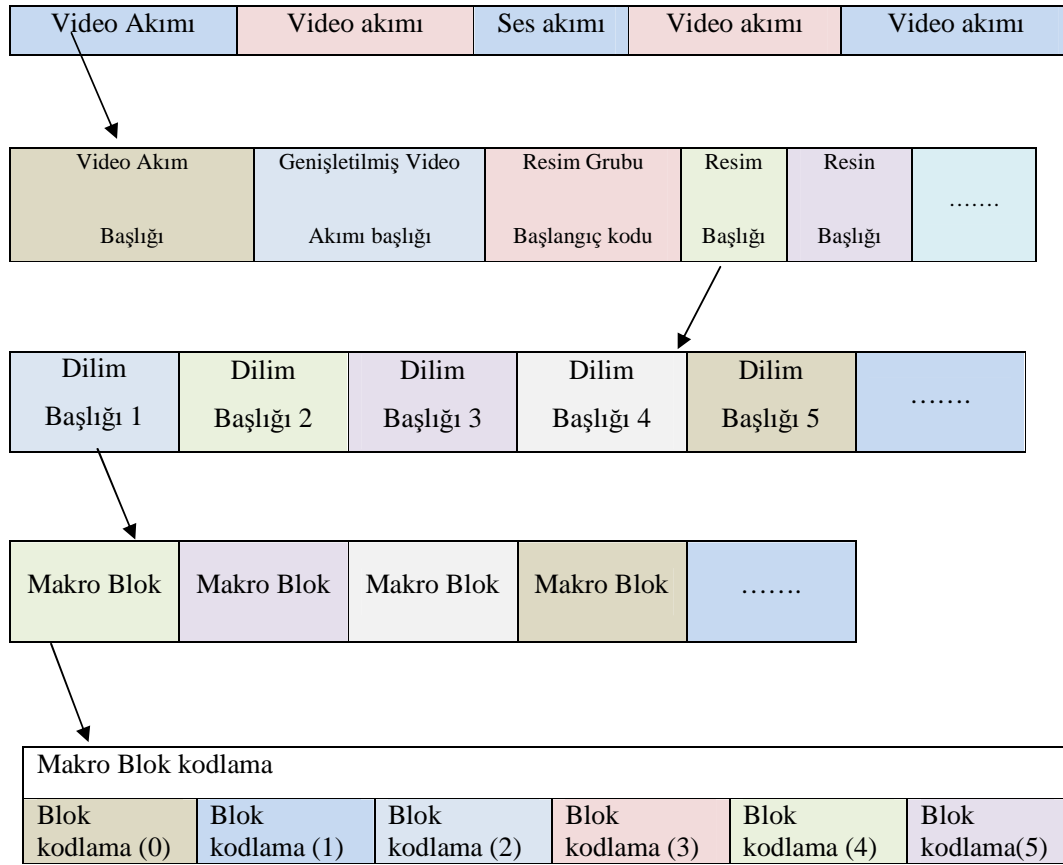
00	0
01	1
10	2
110	3
1110	4
1111 0	5
1111 10	6
1111 110	7
1111 1110	8
1111 1111 0	9
1111 1111 10	10
1111 1111 11	11

Sonuç olarak, MPEG-2 sıkıştırma yöntemi içersinde yer alan birden çok başlık bilgisi kullanılarak, sıkıştırılmış dosya içersindeki video ve ses bilgilerini içeren

bölümler birbirinden ayırt edilmektedir. Ayrıca yüksek sıkıştırma yapabilmek için MPEG-2 tarafından kullanılan yapılar da yine başlık bilgileri kullanılarak birbirinden ayırt edilmektedir. Yapılara ait olan bilgiler birbiri içersine geçmiş olarak bulunmakta ve çoğu zaman sıkıştırma oranını arttırmak için değişken uzunlukta kodlama kullanılmaktadır.

## 5. MPEG-2 YÖNTEMİ KULLANILARAK SIKIŞTIRILMIŞ DOSYA İÇERSİNDEN İ-ÇERÇEVELERİNİN ELDE EDİLMESİ

MPEG-2 hareketli video görüntüleri üzerinde büyük ölçüde sıkıştırma yapabilen bir yöntemdir. Özellikle hareketli video görüntüsünün bir noktadan diğer bir noktaya iletimi sırasında sıkıştırılmadan iletimi oldukça maliyetlidir. Bu sıkıştırma yöntemi içerisinde tanımlanmış olan birden fazla yapı bulunmaktadır. Şekil 5.1’de MPEG-2 sıkıştırma yönteminde kullanılan yapılar gösterilmektedir.



Şekil 5.1. MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya yapısı

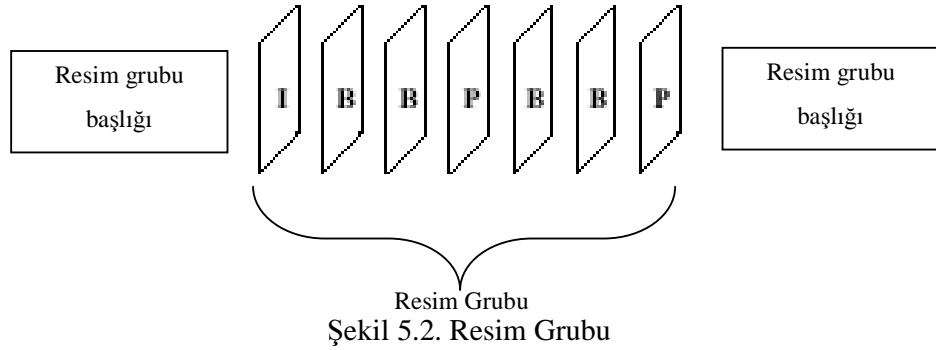
MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içerisinde video akımı ve ses akımı aynı anda bulunabileceği gibi, birden fazla video ve ses akımı da tek bir dosya içerisinde yer alabilmekte ve program paketleri olarak bir noktadan diğerine iletilebilmektedir. Her bir video akımı, video akımı başlığı ile başlamaktadır. Dosyayı

sıkıştırırken MPEG-2 yöntemi kullanıldığında, video akımı başlığından hemen sonra genişletilmiş video akımı başlığı gelmektedir. Genişletilmiş video akımı başlığını resim grubu başlıkları takip etmektedir. Resim grubu birden fazla I, B ve P çerçevelerini içermektedir. Resim grubunun içerdiği her bir resim çerçevesi için ayrı bir resim başlığı bulunmaktadır. Her bir resim çerçevesi dilimlere ayrılmaktadır. Her bir dilim için, dosya içerisinde bir dilim başlığı yer almaktadır. Dosyanın dilim başlıkları arasında kalan bölümlerinde, makro blok ve blokları kodlamak için gerekli olan bilgiler yer almaktadır.

Bu tez çalışmasında, önerilen Video OCR Sistemi, MPEG-2 yöntemi kullanılarak sıkıştırılmış olan hareketli video görüntüsünü içeren dosyada yer alan I-çerçevelerini çözümleyerek bu resim çerçeveleri üzerinde işlem yapacaktır.

### 5.1. Sıkıştırılmış Dosya İçersinde Yer Alan I-Çerçevelerinin Tespiti

MPEG-2 yöntemi çerçeve içi ve çerçeveler arası kodlamanın etkinliğini ve sıkıştırma oranını arttırmak için resim çerçevelerini gruplamaktadır. I-çerçeveleri bu resim grupları içerisinde yer almaktadır.



Şekil 5.2’de resim grubu ve bu resim grubunu oluşturan resim çerçeveleri görülmektedir. Resim grubu başlığı içerisinde, bu resim grubunun hareketli video görüntüsü içerisinde, hangi anda gösterileceği ile ilgili bilgi bulunmaktadır (kaçıncı saat, kaçınıcı dakika, kaçınıcı saniye). Resim çerçevelerini birbirinden ayırt etmek için ise

resim başlığına bakmak yeterlidir. Resim başlığına bakılarak, resim çerçevesinin türü ve resim çerçevesinin geçici referans numarası öğrenilmektedir.

Resim grup başlığını tespit etmek için dosya içerisinde resim grup başlangıç kodunun aranması gerekmektedir. Resim grup başlangıç kodu 0, 0, 1, 184 değerlerini içeren 4 Byte'tan oluşmaktadır.

```

public class mpegDecoder
{
    public byte[] mpegData = null;
    public mpegDecoder(string fileName)
    {
        BinaryReader freader=new BinaryReader(new
        FileStream(fileName, FileMode.Open, FileAccess.Read));
        mpegData = new byte[freader.BaseStream.Length];
        freader.Read(mpegData, 0,(int)freader.BaseStream.Length);
        freader.Close();
    }
    public List<int> findHeaderPosition(int startPosition,List<int> header)
    {
        int i = startPosition;
        List<int> result = new List<int>();
        Boolean headerFound = true;
        while (i < mpegData.Length - header.Count)
        {
            headerFound = true;
            for (int j = 0; j < header.Count;j++)
            {
                if (mpegData[i + j] != header[j])
                { headerFound = false; break; }
            }
            if (headerFound)
            {
                result.Add(i);
                i += header.Count;
            }
            else{
                i = i + 1;
            }
        }
        return result;
    }
}

```

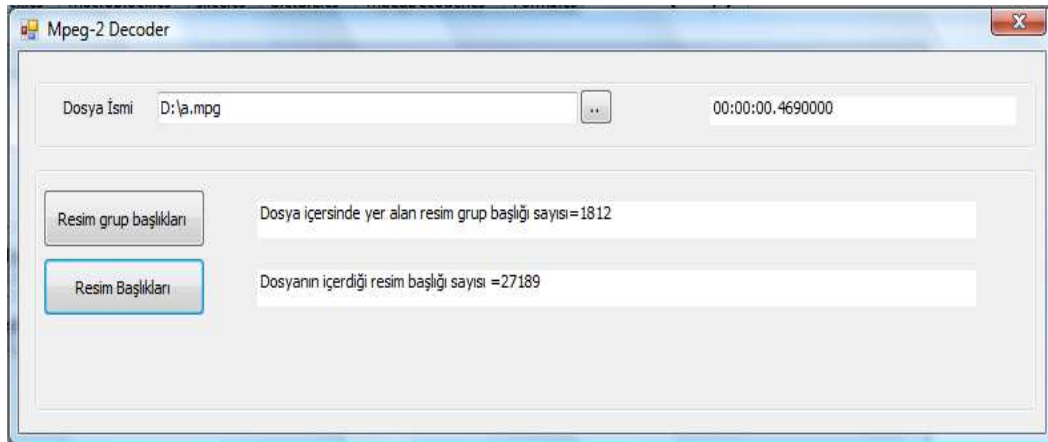
Şekil 5.3. MPEG-2 yöntemiyle sıkıştırılmış dosyalar üzerinde başlık bilgisi bulan C# sınıfı

Şekil 5.3'de MPEG-2 yöntemi kullanılarak sıkıştırılmış olan dosya içerisinde, istenilen başlık bilgisinin konumlarını geri değer olarak döndüren C# programlama dilinde tanımlanan sınıf görülmektedir. Sınıfın yapıcı fonksiyonu, parametre olarak

metin tipinde fileName adında bir değişkene ihtiyaç duymaktadır. Bu parametre ile işlenecek olan dosyanın yolu sınıfa aktarılmış olmaktadır. Daha sonra yapıcı fonksiyon, dosyanın tüm içeriğini, sınıfın genel erişime açık olan mpegData adlı Byte dizisine aktarmaktadır. 295 MB boyutunda bir sıkıştırılmış dosyanın Byte dizisine aktarılması işlemi yarım saniyenin altında bir sürede gerçekleşmektedir.

Resim grup başlangıç kodları ve resim başlangıç kodlarının konumlarını bulmak için **findHeaderPosition (int startPosition, List<int> header)** adlı fonksiyona aramanın dosyanın hangi konumundan başlanacağını belirten bir tamsayı değeri ve bir tamsayı listesini parametre olarak göndermek gerekmektedir. Resim grup başlangıç kodlarının konumlarını bulmak için gönderilen tamsayısı listesi **0, 0, 1, 184** değerlerinden oluşmaktadır. Bu fonksiyon çalıştığında geri dönüş değeri olarak, resim grup başlangıç kodlarının dosyanın hangi konumlarında yer aldığı bilgisini içeren bir tamsayı listesi döndürmektedir. Aynı şekilde, parametre olarak **0, 0, 1, 0** değerlerinden oluşan bir tamsayı listesi fonksiyona iletildiğinde, dosya içersinde resim başlangıç kodlarının konumlarını içeren bir tamsayı listesi sonuç olarak üretilmektedir.

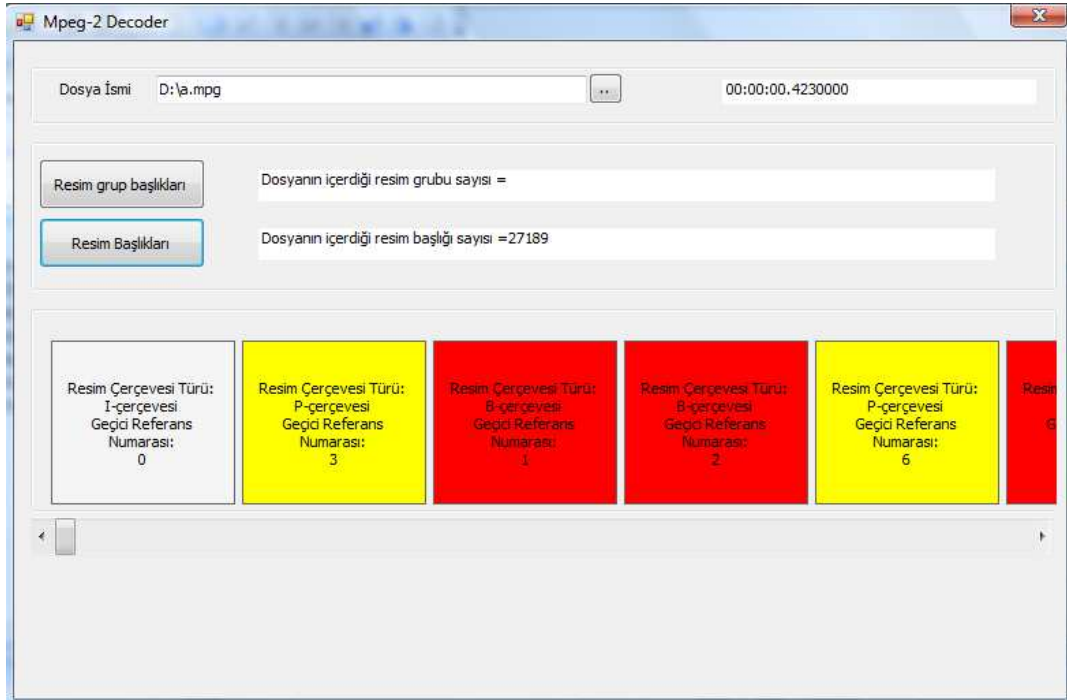
295 MB'dan oluşan örnek dosya için 1812 adet resim grubu başlangıç kodu ve 27.189 adet resim başlangıç kodu tespit edilmektedir.



Şekil 5.4. MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içersinde başlık bilgilerinin konumunu ve sayısını bulan örnek program



Şekil 5.4’de MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içerisinde başlık bilgilerinin konumunu ve sayısını bulan örnek programın ara yüzü görülmektedir. MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içersinden sadece I-çerçevelerini elde etmek için ilk olarak resim gruplarının ve resim çerçevelerinin dosya içersinde bulunduğu konumlar tespit edilmelidir. Resim başlangıç kodunun hemen ardından gelen 2 Byte’a bakılarak, resim çerçevesinin geçici referans numarası ve resim çerçevesinin tipi öğrenilebilir. Ancak 2 Byte 16 Bitten oluşurken, geçici referans numarası değeri bu 2 Byte’ın soldan ilk 10 Biti, onluk sisteme çevrilerek elde edilmektedir. Bu 10 Bitten sonra gelen 3 Bite bakılarak resim çerçevesinin tipi belirlenmektedir.



Şekil 5.5. Çerçeve tiplerini ve çerçeve geçici referans numaralarını bulan örnek uygulama

Şekil 5.5’de çerçeve tiplerini ve geçici referans numaralarını bulan örnek uygulama görülmektedir. Her bir resim çerçevesi türünü belirlemek için, **getPictureInfo(int startPosition)** adlı fonksiyon kullanılmaktadır. Şekil 5.6’da **getPicture(int startPosition)** adlı fonksiyonun içeriği görülmektedir. >> operatörü ile

Bit düzeyinde sağa kaydırma işlemi yapılarak resmin geçici referans numarası değeri elde edilirken, << operatörü kullanılarak Bit düzeyinde sola kaydırma işlemi yapılarak resim çerçevesi tipi belirlenmektedir.

```

public picture getPictureInfo(int startPosition)
{
    picture result = new picture(0);
    byte[] temp=new byte[]{0,0,mpegData
                        [startPosition+4],mpegData[startPosition+5]}
        ;
    Array.Reverse(temp);

    int temporalReference = BitConverter.ToInt32(temp,0);
    result.temporalReference = temporalReference>>6;
    result.pictureType = (temporalReference << 26) >> 29;
    switch (result.pictureType)
    {
        case 1:
            result.pictureTypeName = "I-çerçevesi";
            break;
        case 2:
            result.pictureTypeName = "P-çerçevesi";
            break;
        case 3:
            result.pictureTypeName = "B-çerçevesi";
            break;
    }

    return result;
}

```

Şekil 5.6. Dosya içerisindeki konumu verilmiş olan resim çerçevesi ile bilgi döndüren fonksiyon

## 5.2. I-Çerçevelerini Oluşturan Dilimlerin Tespiti

MPEG-2 yöntemi kullanılarak sıkıştırılmış dosya içerisinde yer alan I-çerçevelerinin konumları tespit edildikten sonra, bu resim çerçevelerini oluşturan dilimlerin konumlarını bulmak ve dilimler üzerinde makro blok çözme işlemi yapmak gerekmektedir.

Dilim başlangıç kodu 4 Byte'tan oluşmaktadır. Diğer başlangıç kodlarından farklı olarak, 175 farklı dilim başlangıç kodu bulunmaktadır. Dilim başlangıç kodlarının

ilk 3 Byte'ı sırasıyla 0, 0, 1 değerlerinden oluşurken, son Byte'ı 1 ile 175 arası değerler alabilmektedir. Dilim başlangıç kodunun bu son Byte'ı dilim içerisinde tanımlanmış olan ilk makro bloğun dikey olarak resim içerisindeki konumunu belirlemek için kullanılır.



Şekil 5.7. Örnek Resim Çerçevesi

Her bir dilim genişliği makro blok genişliği ile aynıdır (16 piksel). Şekil 5.7'de görülen örnek resim çerçevesi içerisinde tanımlanmış olan örnek dilim için, dilim başlangıç kodunun son Byte'ı 3 değerini alacaktır ve bu dilimin örnek resmin sol üst köşesinden olan uzaklığı makro blok yükseklik değeri olan 16 ile bu 3 değerinin 1 eksiğinin çarpımına eşittir. Dolayısıyla, bu örnek resim içerisinde yer alan örnek dilimin, örnek resmin sol üst köşesinden olan uzaklığı 32 piksel olmaktadır.

Dilim başlığında, dilim başlangıç kodundan sonra gelen ilk Byte'ın ilk 5 Biti **nicelendirme ölçüt kodu** bilgisini içermektedir. Nicelendirme ölçüt koduna bakılarak, nicelendirme ölçüt değeri elde edilmektedir. Ancak, nicelendirme ölçüt koduna referans olarak 2 farklı tablo kullanılmaktadır ve genişletilmiş resim başlığında nicelendirme ölçüt değerini belirlerken bu tablolardan hangisinin kullanılacağı **nicelendirme ölçüt**

**tipi değişkeni** ile belirlenmektedir. Nicelendirme ölçüt değeri, sıkıştırılmış resim çerçevesinin çözümü sırasında kullanılacak önemli bir değerdir.

Nicelendirme ölçüt kodunun (5 Bit) hemen ardından gelen 1 Bit 0 değerini alıyorsa, makro blok kodlama işlemine geçilmektedir. Eğer nicelendirme ölçüt kodunun hemen ardından gelen 1 Bit 1 değerini alıyorsa, bu çerçeve içi kodlanmış dilim işaretinin ve başlık içersinde ayrılmış Bitlerin olduğunu ifade etmektedir. Çizelge 4.11’de yer alan dilim başlığı detaylarına bakıldığında, dilim içersinde ekstra bilgilerin de bulunabileceği görülmektedir.

```

while (i<result.endPosition)
{
    if (mpegData[i]==0 & mpegData[i+1]==0 & mpegData[i+2]==1)
    {
        if (mpegData[i+3]>0 & mpegData[i+3]<175)
        {
            slice s=new slice();
            s.sliceStartPosition=i;
            s.verticalPosition = mpegData[i + 3];
            s.vericalPositionPixel = mpegData[i + 3] * 16;
            i += 4;
            while (true)
            {
                if (mpegData[i+4]==0 & mpegData[i+5]==0
                    & mpegData[i+6]==1)
                {
                    break;
                }
                i++;
            }
            s.sliceEndPosition=i+3;
            result.slices.Add(s);
        }
        i++;
    }
}

```

Şekil 5.8. Resim çerçevelerini oluşturan dilimlerin başlangıç ve bitiş konumlarını tespit eden C# kodu

Şekil 5.8’de resim çerçevelerini oluşturan dilimlerin başlangıç ve bitiş konumlarını tespit eden C# kodu görülmektedir. Başlangıç ve bitiş konumları tespit

edilen dilimler ise dilimleri kodlamak için kullanılan **decodeSlice()** fonksiyonuna yönlendirilmektedir.

```

public void decodeSlice(slice s)
{
    int i = s.sliceStartPosition;
    s.quantiserScaleCode = mpegData[i + 4] >> 3;

    if (((byte)(mpegData[i + 4] << 5) >> 7) != 0)
    {
        s.intraSliceFlag = 1;
        s.intraSlice=((byte)(mpegData[i + 4] << 6) >> 7);
    }
    else //extra bit slice=0
    {
        decodeMacroBlocks();
    }
}

```

Şekil 5.9. Dilimleri kodlamak için kullanılan C# kodu

Şekil 5.9’da dilimleri kodlamak için kullanılan C# kodu görülmektedir. Her bir makro bloğun kodlaması Bit düzeyinde yapılmaktadır. Dilim başlığından başlayarak, bir sonraki dilim başlığına kadar, dilim içersinde yer alan tüm makro bloklar **decodeMacroBlocks()** fonksiyonu ile çözümlenmektedir.

### 5.3. Makro Blok Kodlama

Makro blok kodlama işleminin ilk adımında makro blok adres artım değeri bulunmaktadır. **Makro blok adres artım değeri**, kodlanmakta olan makro blok ile bu makro bloktan önce kodlanmış olan makro bloğun yatay konumu arasındaki fark değeridir. Dilimin en solunda yer alan makro bloğun adresi 0 olarak kabul edilmektedir. Makro blok adres artım değerinin 33’ten fazla olduğu durumlarda ise **makro blok kaçış kodu** kullanılmaktadır. Makro blok kaçış kodu, 11 Bitten oluşmaktadır ve “0000 0001 000” değerine sahiptir. Makro blok adres artım miktarının 33’ten fazla olduğu durumlarda sıkıştırılmış dosya içersinde yer alan bu kod, makro blok adres artım değerinin 33’e tam bölümünden elde edilen sonuç kadar tekrarlanmaktadır. Örneğin makro blok adres artım değerinin 85 olduğu durumda, sıkıştırılmış dosya içersinde bu

adres artım deęerini kodlamak için  $85 \setminus 33 = 2$  makro blok kaçış kodu yan yana yer almaktadır. 85 deęerinden  $66(33 \times 2)$  deęerinin çıkarılmasından elde edilen 19 deęeri ise deęişken uzunlukta kodlama yöntemi ile kodlanmaktadır. Çizelge 4.13'de yer alan tablo kullanılarak makro blok adres artım deęeri kodlanmaktadır. I-çerçevesini çözerken, sıkıştırılmış dosya içersinde kodlanmayan makro blok olmadığı için makro blok kaçış koduna rastlanmadığı gibi, makro blok adres artım miktarı da sürekli olarak 1 deęerini almaktadır.

Makro blok kodlamanın ikinci adımında, kodlanacak olan makro blok tipi belirlenmektedir. Makro blok tipi belirleme için **makro blok tip belirle()** alt yordamı kullanılmaktadır. I-çerçeveleri kodlanırken **çerçeve içi kodlanmış makro blok** ile **nicelendirme ölçütlü çerçeve içi kodlanmış makro blok** olmak üzere 2 farklı türde makro blok kullanılmaktadır.

Makro blok tipinin nicelendirme ölçütlü çerçeve içi kodlanmış olması durumunda, makro blok adres artım deęerini kodlamak için kodlanan Bitlerin hemen ardından 5 Bitten oluşan nicelendirme ölçüt kodu yer almaktadır. Nicelendirme ölçüt koduna bakılarak çerçeve içi kodlama yönteminin bir aşaması olan nicelendirme işlemi sırasında kullanılan parametre elde edilmektedir. Makro blok tipi, çerçeve içi kodlanmış makro blok ise nicelendirme ölçüt kodu olarak dilim başlığında yer alan deęer kullanılmaktadır.

Çerçeve içi kodlanmış bir makro blok çözümlenirken, makro blok tipi belirleme fonksiyonundan sonra blok kodlama işlemine geçilmektedir.

#### 5.4. Blok Kodlama

Bloklar, MPEG-2 sıkıştırma yönteminde kullanılan yapılar içersinde en alt seviyede yer almaktadır. Dilimleri oluşturan makro bloklar, renk bilgisi alt örnekleme türüne baęlı olarak, 4 veya daha fazla bloktan oluşmaktadır. Genişletilmiş resim başlığında yer alan **chroma 420 tipi** işaretime bakılarak, makro blokların 4:2:0 renk

bilgisi alt örnekleme türünü kullanarak kodlandığı tespit edilmektedir. Bu makro blok, 4 parlaklık bilgisi ve 2 renk bilgisi (mavi ve kırmızı) içeren 6 bloktan oluşmaktadır.

Blokları kodlarken büyük önem taşıyan çerçeve içi kodlama değişken uzunlukta kodlama format değeri genişletilmiş resim başlığı içersinde yer almaktadır ve değişken uzunlukta kodlama yapılırken kullanılacak tabloların belirlenmesi için kullanılmaktadır.

Çizelge 5.1. Değişken uzunlukta kodlama için kullanılacak tabloların seçimi

Değişken uzunlukta kodlama format değeri	0	1
Çerçeve içi kodlama yapılırken	Tablo B-14	Tablo B-15
Çerçeveler arası kodlama yapılırken	Tablo B-14	Tablo B-14

Çizelge 5.1'deki değişken uzunlukta kodlama format değerine bakılarak, çerçeve içi kodlama ve çerçeveler arası kodlama kullanılarak sıkıştırılmış olan resim çerçevelerini çözerken, değişken uzunlukta kodlama için hangi tablonun kullanılacağı belirlenmektedir.

Blok kodlama işlemi sonucunda, 8x8 boyutundaki bloklara ait parlaklık ve renk bilgisi elde edilmektedir. Bu bloğun (0,0) konumunda yer alan ilk elemanına **DC katsayısı**, diğer 63 elemanına ise **AC katsayıları** adı verilmektedir. DC katsayısı ve AC katsayılarının çözümlenmesi sırasında farklı yöntemler kullanılmaktadır. Çizelge 4.14'de blok kodlama yordamının detayları yer almaktadır. Parlaklık bilgisi içeren bloklar ve renk bilgisi içeren blokları çözümlerken farklı yöntem izlenmektedir.

DC katsayısı değerini elde etmek için, değişken uzunlukta kodlanmış DC katsayısı fark değerinin Çizelge 4.15 ve Çizelge 4.16'ya bakılarak kaç Bit kullanılarak kodlandığı tespit edilmektedir. Daha sonra elde edilen bu Bit sayısı değeri dikkate alınarak DC katsayısı fark değeri elde edilmektedir.

DC katsayısı değeri elde edildikten sonra, 8x8 boyutundaki bloğunda diğer 63 elemanın değeri çözümlenerek elde edilmektedir. Blok kodlama işlemi, değişken

uzunlukta kodlama yapılırken, Tablo B-14 ve Tablo B-15’de yer alan ve **Blok sonu**’nu ifade etmek için kullanılan Bit dizisi ile karşılaşıncaya kadar devam etmektedir.

Blok kodlama sırasında kullanılan değişken uzunlukta kodlama yönteminde, blok dizisinin birden fazla elemanı, tek bir işlemde kodlanabilmektedir. Kodlama sırasında, 3 adet değişken kullanılmaktadır. Bu değişkenler **ilerleme**, **düzey** ve **işaretli düzey** değişkenleridir. Çözümleme sırasında ilk olarak ilerleme değişkeninin değeri belirlenmektedir. İlerleme değişkeninin değeri belirlendikten sonra ise düzey ve işaretli düzey değişkenleri belirlenmektedir.

İlerleme değişkeni, kodlamaya başlanılan noktadan sonra gelen ve 0 değerinden farklı bir değere rastlanıncaya kadar 0 değerine sahip dizi elemanlarının sayısını ifade etmektedir. Örneğin 0, 0, 0, 0, -2, 3, 0, 0, -4 elemanlarından oluşan bir dizi baştan başlayarak kodlandığında, ilk kodlama kelimesinin ilerleme değişkeni değeri 4 olmaktadır. İşaretli düzey değişkeninin değeri -2’dir. Bu kodlama adımı için düzey değişkeninin değeri ise işaretli düzey değişkeninin mutlak değeri olan 2 olmaktadır.

Kodlamanın ilk adımında 0, 0, 0, 0, -2 değerlerine sahip ilk 5 eleman kodlanmıştır. Bu değer dizisi 3 değişken kullanılarak kodlanmıştır. Bu değişkenlerin değerlerini kodlayarak Bit dizisine dönüştürmek için referans tablolar olan Tablo B-14 ya da Tablo B-15 kullanılmaktadır. Tablo B-15’in referans olduğu varsayılarak bu 3 değişken “11111011” şeklinde kodlanmaktadır. Değişken uzunlukta kodlamanın kullanılmadığı durumda, 5 elemandan oluşan bu dizi sayısal ortamda  $5 \times 8 = 40$  Bit kullanılarak saklanabilmektedir. Çerçeve içi kodlama yönteminde yer alan değişken uzunlukta kodlama kullanılarak ise bu 5 elemandan oluşan diziyi sayısal ortamda saklamak için 9 Bit’e ihtiyaç vardır.

Bu örnek dizinin kodlanmasına devam edildiğinde ikinci adımda, ilerleme değişkeni 0, düzey değişkeni 3, işaretli düzey değişkeni 3 değerini almaktadır. Tablo B-15’e bakıldığında bu değişkenleri kodlamak için “01110” Bit dizisi kullanılmaktadır. Örnek dizinin 3. Adımında ise ilerleme değişkeni 2, düzey değişkeni 4, işaretli düzey



değişkeni ise -4 değerini almaktadır. Tablo B-15'e bakıldığında bu değişkenleri kodlamak için "00000011001" Bit dizisi kullanılmaktadır. Örnek dizinin tüm değişkenlerinin değişken uzunlukta kodlama yöntemi kullanılarak kodlanması sonucunda "111110110111000000011001" Bit dizisi elde edilmektedir. 10 adet elemandan oluşan örnek diziyi kodlamak için 25 Bit uzunluğunda Bit dizisi kullanılmıştır. 25 Biti sayısal ortamda saklamak için 4 Byte gerekmektedir. Örnek Bit dizisini sıkıştırma yapmadan saklamak için ise, 10 Byte gerekecektir.

Blokların AC değişkenleri kodlanırken, yukarıda anlatılan yöntem kullanılmaktadır. Çözümlemlenirken de bu yöntemin tersi adımlar takip edilmektedir. Şekil 5.10'da örnek resim çerçevesinin 1. dilimini kodlayan Bit dizisi görülmektedir. Bu Bit dizisi dilimi oluşturan tüm makro blokları ve makro blokları oluşturan blokları kodlamaktadır.

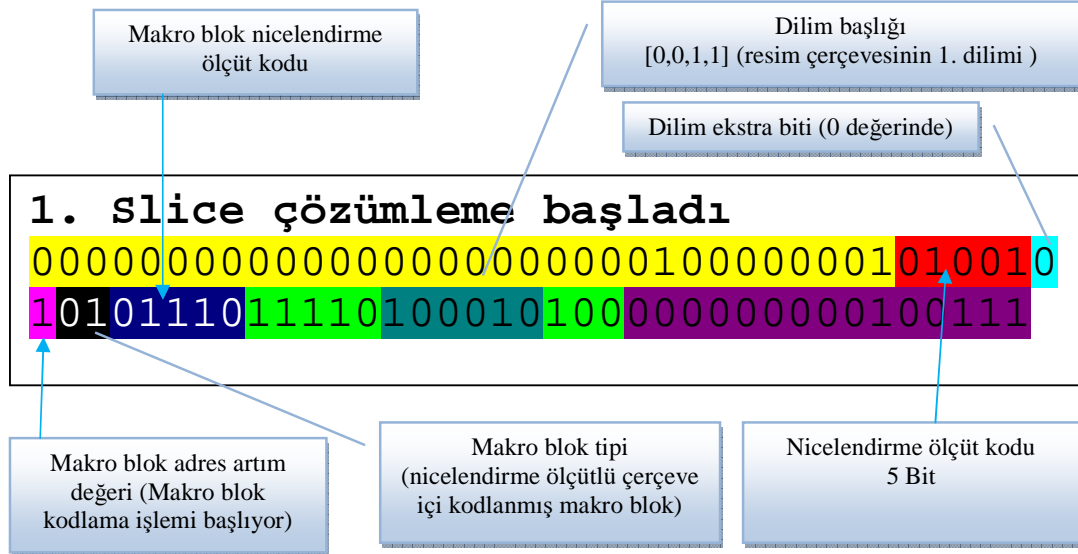
```

1. Slice çözümleme başladı
00000000000000000000000100000001010010101011101110100010100000011111111111001000110000000100010
01001110000000001111010100000001000011111111111001000110000000100010
1000000000100111010000001001101000000000110110111110001110001110001
00010100010111101011100111010100110100111111001100101001110110000011
0110000001000011011111010011000110001101111100000010001000110101010
1111101100001010101000111111000110000100001001101101001101000010110
110100010001101010110111100000000000000000000111001000100100000000100101
00000000100000110000100100001100010000000000111001000100100000000100
1011110100110000100100001101101111011001100110101101010000110101101
011011011111001101000000000111001000100100000001001011111010011000
0100100001100000000000000111001000101100000001001011111010011000010
01000011011101011001100000110010001011000111101100110110101011001110
01001000000000011011100010010000000010010100000000100000110000100100
001100000000000001101010001001000001001111101001100001001000011011
1010110011010001100100010110001111011001101010110111100111011011111
1011111101111011000010100111101000000000100010110000100101110001101
1101100111010000000001110111111010111001010100111100111010000000010
0010111000000011000000001111000110101011011011101001100010011100001
11000001011011010111010011010001111011011001011001011000001101011011
1100100000001000001111111010110010001010000010001011111111111001111
01001110000000000000011000001100000000010000011111111010110010001110
00000001001011001111000001110000000000001100000110111011101001100000
00001111100000100101100110010010100001101111101000110110110111100111
10101101110010001001100011010101011111000001000001000001111111100111
111110101000001000101111111111001000000000100010011100000110000000
0000000101000011010100110
1. Slice çözümleme bitti

```

Şekil 5.10. Örnek resim çerçevesinin 1. dilimini kodlayan Bit dizisi

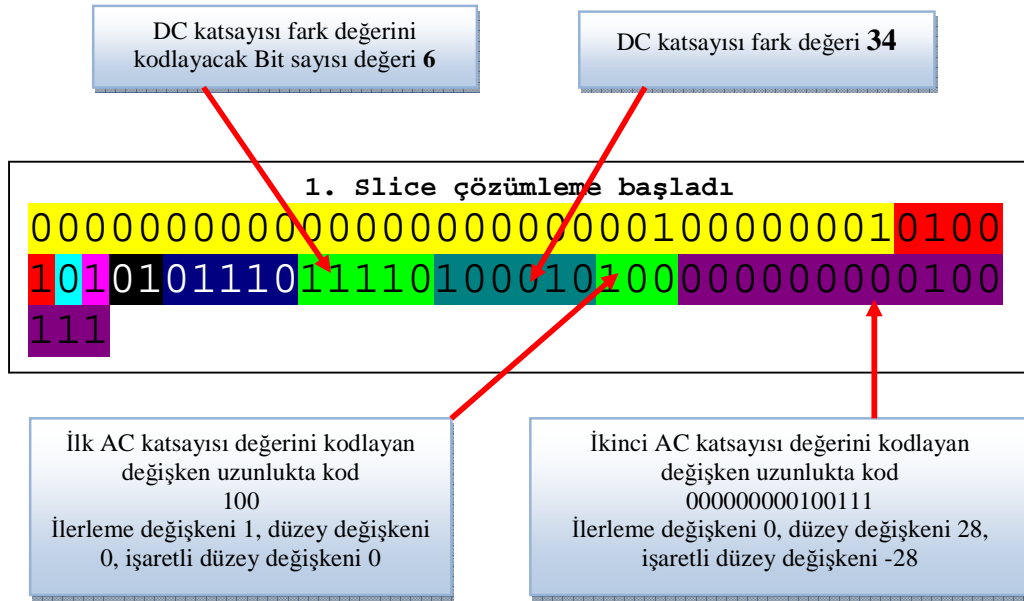
Şekil 5.10'da yer alan Bit dizisi çözümlenirken ilk olarak dilim başlığı ile karşılaşılmaktadır. Dilim başlığı çözümlendikten sonra makro blok kodlama yapılmaktadır. Makro blok kodlama, **makro blok tipi belirle()** ve **blok kodlama()** alt yordamlarını çalıştırarak, blokların matris değerlerini elde etmektedir.



Şekil 5.11. Örnek resim çerçevesine ait 1.dilimin çözümme aşamaları – Dilim başlığı ve makro blok kodlama

Şekil 5.11'de örnek resim çerçevesine ait 1. dilimin çözümleme aşamalarından dilim başlığı ve dilimi oluşturan 1. makro bloğun kodlama işlemi detaylandırılmaktadır. Sonraki aşamalarda makro bloğu oluşturan bloklar kodlanacaktır.

Şekil 5.12'de blok kodlama aşamaları görülmektedir. Blok sonu kodunu temsil eden Bit dizisi ile karşılaşıncaya kadar blok kodlama işlemi devam etmektedir.



Şekil 5.12. Örnek resim çerçevesine ait 1.dilimin çözülme aşamaları – Blok kodlama

Şekil 5.12’de örnek resim çerçevesinin 1. diliminin 1. makro bloğunu oluşturan ilk bloğun çözümlendiği adımlar görülmektedir. Bu bloğun DC katsayısı değeri 34 tür Şekilde görülen Bit dizisini çözmeye devam ettiğimizde ise AC katsayılarından 4’ü elde edilmektedir. Bu 4 değer 0, 1, 0, 28’dir. Blok kodlama işleminin tersi olan blok çözme işlemi bu şekilde yapılmaktadır. Blok sonu kodu ile karşılaşıldığında blok kodlama işlemi bitmektedir. Bu kod ile karşılaşıldığında, bloğun tüm AC katsayı değerleri çözülmemiş olabilir. Blok sonu kodu ile karşılaşılan kadar değeri tespit edilememiş olan tüm değerlerin 0 olduğu kabul edilmektedir.

Blok kodlama işlemi sırasında değişken uzunlukta kodlama yapıldığından bloklar çözülürken 1 Bitin değerinin farklı olması bile bloğun düzgün bir biçimde çözümlenmesini engellemektedir.

## 5.5. Video Çözme İşlemleri

MPEG-2 yöntemi kullanılarak sıkıştırılmış hareketli video dosyası içersinden I-çerçevelerinin elde edilmesi sırasında, blok kodlama işleminden başlayarak devam eden adımlara **video çözme işlemleri** adı verilmektedir.

I-çerçeveleri elde edilirken video çözme işleminin içerdiği adımlar aşağıda listelenmektedir;

1. Değişken uzunlukta kodlanmış Bit dizisini çözümleyerek 64 Byte uzunluğunda Byte dizisi elde etme
2. Elde edilen Byte dizisi üzerinde
  - a. Ters tarama
  - b. Ters nicelendirme
  - c. Ters ayrık kosinüs dönüşümü
  - d. YUV renk uzayından RGB renk uzayına dönüşüm

Yukarıda yer alan adımlarlardan ilki, **blok kodlama()** yordamı ve bu yordama gelene kadar gerçekleştirilen çözme aşamalarını içermektedir. Bu aşamadan sonra blokları oluşturan 8x8 boyutundaki piksellerin R, G, B renk değerlerini tekrar elde etmek için ilk olarak ters tarama işlemi yapılmaktadır. Ters tarama işlemi kodlama aşamalarında anlatılan zigzag taramanın tersidir. Zigzag tarama kullanılarak, nicelendirme işlemine tabi tutulmuş olan 8x8 boyutundaki matris, tek boyutlu bir dizi haline dönüştürülmekte iken, ters tarama işleminde ise 64 elemandan oluşan tek boyutlu Byte dizisi, 8x8 boyutunda bir matrise dönüştürülmektedir. Ters tarama işleminde zigzag tarama işleminde olduğu gibi, tarama desenleri kullanılmaktadır. Kodlama sırasında, hangi tarama deseni kullanıldı ise, ters kodlama sırasında da bu desenin kullanılması gerekmektedir. Şekil 3.14’de ters tarama işlemi sırasında kullanılan tarama desenleri görülmektedir.

Şekil 5.13’de ters tarama işlemi yapan program kodu görülmektedir. Genişletilmiş resim başlığı içerisinde yer alan alternatif tarama değişkenin almış olduğu değer dikkate alınarak, tek boyutlu bir diziden 2 boyutlu bir matris elde edilmektedir.

```
for (int v=0;v<8;v++)
  for (int u=0;u<8;u++)
    QF[v,u]=OFS[tarama deseni[alternatif tarama,v,u]]
```

Şekil 5.13. Ters tarama işlemi

Video çözme işleminin her adımı, resim çerçevesinin video sıkıştırma işlemi yapılmadan önceki halini elde etmeye yöneliktir ve ters nicelendirme aşaması dışında kalan aşamalar için sıkıştırma sırasında kayıp olmamaktadır. Video görüntüsü kodlanırken, nicelendirme işlemi sırasında bölme ve yuvarlama işlemi yapıldığından, insan gözünün fark edemeyeceği düzeyde bir kayıp söz konusu olmaktadır. Ters nicelendirme işlemi basit bir çarpma işleminden oluşmaktadır.

Ters nicelendirme işlemi yapılırken, kullanılan önemli bir parametre de nicelendirme ölçüt kodudur. Nicelendirme ölçüt kodu her dilim başlığında yer almaktadır. Eğer makro blok kodlama işlemi yapılırken, ilk kodlanan makro blok nicelendirme ölçüt kodu içeren bir makro blok değilse, dilim başlığında yer alan nicelendirme ölçüt kullanılmaktadır. Kodlanan makro blok nicelendirme ölçütü içeren bir makro blok değil ise, ters nicelendirme işlemi sırasında, bu makro bloktan önce gelen makro bloğun kodlarken kullanılan nicelendirme ölçütü kullanılmaktadır. Nicelendirme ölçütünün değerini belirlerken, kullanılan bir diğer parametre de genişletilmiş resim başlığında yer alan nicelendirme ölçüt tipi değişkenidir. Bu değişken ve nicelendirme ölçüt kodu değerine bakılarak ters nicelendirme işleminde kullanılacak olan nicelendirme değeri belirlenmektedir. Çizelge 5.2 nicelendirme ölçüt değerini elde etmek için kullanılmaktadır. Nicelendirme ölçüt kodu 1 ile 31 arası değerler almaktadır. Genişletilmiş resim başlığında yer alan nicelendirme ölçüt tipi değerine bakılarak nicelendirme ölçüt değeri elde edilmektedir.

Ters nicelendirme işlemi sırasında blokların DC katsayısı değeri ile AC katsayısı değerlerinin elde edilmesi sırasında farklı yöntemler kullanılmaktadır. DC katsayısı değeri, genişletilmiş resim başlığında yer alan çerçeve içi DC katsayısı duyarlılık değerine bakılarak elde edilen DC katsayısı çarpanı ile çarpılarak elde edilmektedir. Çizelge 5.3’de çerçeve içi kodlama DC katsayısı duyarlılık değerine karşılık belirlenmiş olan DC katsayısı çarpanı değerleri görülmektedir.

Çizelge 5.2. Nicelendirme ölçüt değeri tablosu

Nicelendirme ölçüt kodu	Nicelendirme ölçütü tipi 0	Nicelendirme ölçütü tipi 1
1	2	1
2	4	2
3	6	3
4	8	4
5	10	5
6	12	6
7	14	7
8	16	8
9	18	10
10	20	12
11	22	14
12	24	16
13	26	18
14	28	20
15	30	22
16	32	24
17	34	28
18	36	32
19	38	36
20	40	40
21	42	44
22	44	48
23	46	52
24	48	56
25	50	64
26	52	72
27	54	80
28	56	88
29	58	96
30	60	104
31	62	112

Çizelge 5.3. Çerçeve içi kodlama DC duyarlılık değeri - DC katsayısı çarpanı değeri tablosu

Çerçeve içi kodlama DC katsayısı duyarlılık değeri	DC katsayısı çarpanı değeri
0	8
1	4
2	2
3	1

AC katsayılarına ait değerleri elde etmek için, nicelendirme ölçüt değeri ve nicelendirme matrisi kullanılmaktadır. Eğer, genişletilmiş akım başlığı içerisinde çerçeve içi kodlama ve çerçeveler arası kodlama için kullanılacak özel nicelendirme matrisleri yok ise varsayılan nicelendirme matrisi kullanılarak AC katsayılarına ait değerler hesaplanmaktadır.

```

for (int v=0;v<8;v++)
{
    for (int u=0;u<8;u++)
    {
        Eğer (u==0) & (v==0) ise
        {
            F'[v,u]=DC çarpanı değeri * QF[v,u];
        }
        değilse
        {
            F'[v,u]= (QF[v,u]; * nicelendirme matrisi [v,u] * nicelendirme
            ölçütü değeri) /32
        }
    }
}

```

Şekil 5.14. DC ve AC katsayılarını elde etmek için kullanılan formül

Şekil 5.14'de DC ve AC katsayılarını elde etmek için kullanılan formül yer almaktadır. Bu kod sonucunda 8x8 boyutunda bir matris elde edilmektedir. Elde edilen matrisi oluşturan elemanların değerlerinin -2048 ile 2047 arasında olması gerekmektedir. Değeri -2048 den küçük olan matris elemanlarının değeri de -2048'e,

2047 den büyük olan matris elemanlarının değeri 2047'ye dönüştürülmektedir. Bu düzenleme işlemine **doygunluk kontrolü** adı verilmektedir. Doygunluk kontrolüne benzeyen ve ters ayrık kosinüs dönüşümünden önce gelen son işlem ise **yanlış eşleştirme** kontrolüdür. Yanlış eşleştirme kontrolünde, ilk olarak doygunluk kontrolü sonucunda elde edilmiş olan tüm değerler toplanır. Eğer elde edilen bu toplam değer çift sayı ise, doygunluk kontrolü sonucunda elde edilmiş olan matrisin son elemanın değerinin çift ya da tek olup olmadığı kontrol edilmektedir. Eğer bu matrisin son elemanın değeri çift ise bu son elemanın değeri 1 artırılmakta, tek ise 1 azaltılmaktadır.

```

toplam=0
for (int v=0;v<8;v++)
{
    for (int u=0;u<8;u++)
    {
        F''[v,u]=F'[v,u]
        Eğer (F'[v,u] >2047) ise F''[v,u]=2047
        Eğer (F'[v,u] <-2048) ise F''[v,u]=-2048
        toplam+=F''[v,u]
    }
}

Eğer (toplam %2==0) ise
{
    Eğer (F''[v,u]%2==0) ise
    {
        F''[v,u]+=1
    }
    Değilse
    {
        F''[v,u]-=1
    }
}

```

Şekil 5.15. Doygunluk ve yanlış eşleştirme kontrolleri

Şekil 5.15'da doygunluk ve yanlış eşleştirme kontrollerini yapan formül görülmektedir. Bu kontroller yapıldıktan sonra matris üzerinde ters ayrık kosinüs dönüşümü yapılacaktır. Ters ayrık kosinüs dönüşümü sonucunda elde edilen değerler blokların YUV uzayındaki renk değerleridir. YUV renk uzayından RGB renk uzayına



dönüşüm formülleri (5-1)'de verilmektedir. Bu değerlerin RGB renk uzayına döndürülmesi ile video çözme işlemi tamamlanmaktadır.

```
int c = Y - 16;  
int d = U - 128;  
int e = V - 128;  
int R = ((298 * c + 409 * e + 128) >> 8) (5-1)  
int G = ((298 * c - 100 * d - 208 * e + 128) >> 8)  
int B = ((298 * c + 516 * d + 128) >> 8)
```

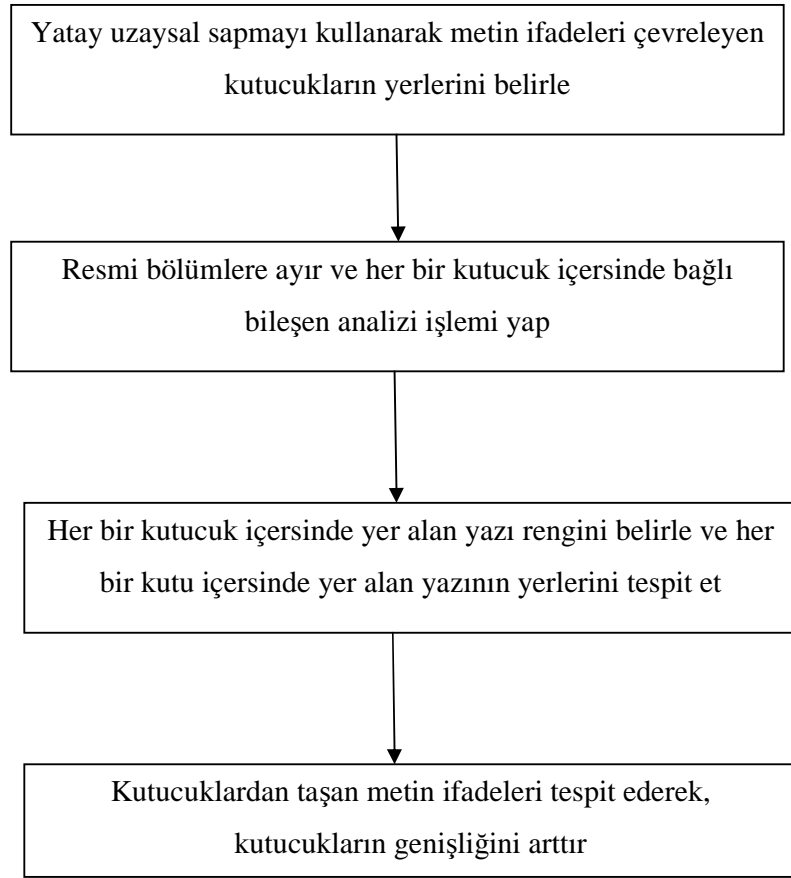
## 6. VIDEO OCR SİSTEMLERİ ÜZERİNE YAPILAN ÇALIŞMALAR

Hareketli video görüntüleri içerisinde yer alan metin ifadeleri tespit ve elde edebilen sistemlere Video OCR Sistemleri adı verilmektedir (Sato v.d., 1999). Hareketli video görüntüleri içerisinde yer alan sahne metni ve yapay metinlerin tespit edilmesi, bu görüntülerin içeriği hakkında sorgulama yapılmasına imkân sağlamaktadır. Çoğu zaman sayısal ortamda hareketli video görüntüleri uygun bir sıkıştırma yöntemi kullanılarak depolanmaktadır. Video OCR Sistemlerinin sahne metinlerini ve yapay metinleri elde etmek için kullandığı yöntemler temel olarak 2 gruba ayrılmaktadır. Birinci grup yöntemde, sıkıştırılmış hareketli video görüntüsü çözülerek elde edilen resim çerçevesi içerisinden sahne metni ve yapay metinleri elde etmektedir. Birinci grup yöntemler, temel olarak bir resim üzerinde yer alan metin ifadeleri tespit eden yöntemlere benzerdir ve sıkıştırılmış video dosyası içerisinden tamamı ile elde edilmiş resim çerçeveleri üzerinde işlem yapmaktadır. İkinci grup yöntemler ise, sahne metinlerini ve yapay metinleri sıkıştırılmış video dosyasını açmadan ve resim çerçevelerini çözmeden elde etmektedir. MPEG1 ve MPEG-2 sıkıştırma metodunda yer alan ve üzerinde ayrık kosinüs dönüşümü yapılmış resim çerçeveleri oluşturan matris katsayıları kullanılarak sahne metinleri ve yapay metinler tespit edilmektedir.

### 6.1. Sıkıştırılmış Video Görüntüsünü Çözerek Elde Edilen Resim Çerçevesi İçerisinden Metin İfadeyi Elde Eden Yöntemler

Sıkıştırılmış hareketli video dosyası içerisinden, video görüntüsünü oluşturan resim çerçeveleri elde edildikten sonra, bu çerçeveler içerisinde yer alan metin ifadeleri tespit edebilmek için birçok çalışma yapılmaktadır. Bu çalışmalarda, metin ifadeyi tespit ederken, bu ifadelerin karakteristik özellikleri dikkate alınarak yerleri tespit edilmekte ve bağlı bileşen analizi yapılarak, resim içerisinden metin ifadelerin elde edilmesi sağlanmaktadır. Metin ifadenin konumu dikey kenar yoğunluğu, dokusu ve kenar yönlendirme sapması dikkate alınarak belirlenebilmektedir.

Resim içersinden metin ifadeleri elde etmek için yapılan ilk çalışma Zhong v.d. tarafından yapılmış olup (Zhong v.d., 1995) CD'ler üzerinde yer alan metin ifadelerin yerinin bulunması ve elde edilmesini amaçlamaktadır. Karışık bir arka plana sahip ve birden çok nesneyi içersinde barındıran bir resim için önceden boyutu, yazı tipi ve rengi belirli olamayan bir metin ifadeyi basit eşikleme algoritmasıyla elde etmek mümkün değildir.





Şekil 6.1. Algoritmanın blok diyagramı

Şekil 6.1'de Zhong v.d. tarafından geliştirilen algoritmanın blok diyagramı bulunmaktadır. Algoritmanın ilk adımında yüksek uzaysal sapma değerine sahip yatay satırlar tespit edilmektedir. Siyah beyaz olarak yazdırılmış bir sayfa düşünüldüğünde, tüm resmi oluşturan her bir yatay satır için uzaysal sapma değeri hesaplandığında, yazıların bulunduğu satırlarda uzaysal sapma değerinin yüksek olduğu, hiçbir yazının

yer almadığı sadece beyaz renkten oluşan satırlarda ise uzaysal sapma değerinin düşük olduğu görülmektedir. Yazıyı çevreleyecek olan kutucukların yatay olarak başladığı ve bittiği noktalar bu sapma değerinde meydana gelen büyük değişiklikler saptanarak belirlenmektedir. Yazıyı çevreleyecek olan kutucukların dikey olarak başladığı ve bittiği noktalar ise **kenar belirleme algoritmaları** kullanılarak belirlenmektedir. Yatay konumu belirlenmiş kutucuklar için, kenar belirleme algoritması, en önemli köşelerin (en solda ve en sağda kalan köşeler) konumunu belirleyerek kutunun sol üst ve sağ alt köşeleri belirlenmiş olmaktadır. Belirlenmiş olan bu kutucuğun boyutlarının çok küçük olduğu durumlarda bu kısımda yazının yer alamayacağı varsayılmaktadır. Sol üst ve sağ alt köşesi belirlenmiş olan ve içersinde metin ifadelerin yer alması muhtemel olan kutucuklar üzerinde bağlı bileşen analizi yapılmaktadır. Metin ifadenin bazı kısımlarının kutucuktan taşabilme ihtimali göz önünde bulundurularak bu durum ile ilgili algoritma içersinde kontroller yer almaktadır.

Smith ve Kanade (Smith ve Kanade, 1995) tarafından yapılan ve ABD Ulusal Bilim Kurumu tarafından desteklenen çalışma da ise resim içersinde yer alan metin ifadelerin yerini tespit etmek için yeni bir yöntem ortaya konulmaktadır. Bu çalışma da **metin ifadeler içeren alanlar**; arka plan rengine karşıt ya da daha koyu renklerde olan, gruplanmış keskin köşelere sahip, yatay dikdörtgen alanlardır tanımlaması yapılmaktadır. İlk olarak dikey köşelerin belirlenmesi için, 3x3 boyutunda yatay farklılıklar filtresi ve bu filtreye uygun ikili eşikleme işlemi tüm resim üzerine uygulanmaktadır. Bu aşamadan sonra filtre sonucu elde edilen bölümler üzerinde kümeleme işlemi yapılmakta ve bağlantısı koparılmış olan bölgeler birbirine bağlanmaktadır. Kümelenmiş bölgelerden içersinde metin ifade olanların tespiti için aşağıda yer alan üç kıstas dikkate alınmaktadır.

- Kümelenmiş bölümlerin en boy oranının 0.75'den büyük olması gerekmektedir. 
- Kümelenmiş bölümlerin doluluk oranının 0.45 den büyük olması gerekmektedir. 
- Kümelenmiş bölümün en az 70 pikselden oluşması gerekmektedir.

Yapılan bu iki çalışma, çok karmaşık arka planların olduğu durumlarda düzgün neticeler üretmemektedir.

Resim içersinde yer alan metin ifadelerin tespiti için 1997 yılında Wu v.d. tarafından geliştirilen yöntemde, karakterlerin arka plandan farklı dokulara sahip olduğu göz önünde bulundurularak, doku bölümlene teknikleri kullanılmaktadır. Yöntemin ilk aşamasında, standart çok kanallı filtreleme doku bölümlene tekniğini kullanılarak doku bölümlene işlemi yapılmaktadır. Resim içersinde yer alan metin ifadelerin tespiti için bu filtreleme işlemi yapıldığında, işlenen resim, gazete taramalarından (düz tek renkli arka plan) farklı olarak karmaşık arka planlar içerirse olumlu sonuçlar alınamamaktadır. Buna karşın yapılan filtreleme işlemi, yöntemin ikinci aşaması olan çip üretim aşamasının odaklanacağı resim parçalarını belirlemiş olmaktadır. Çip üretim aşamasında, aşağıdan yukarı yönde işlem yapılmaktadır ve metin ifadelerin ana hatları belirlenmektedir. Metin ifadelerin ana hatlarını belirleme işlem ise aşağıda yer alan aşamalardan oluşmaktadır;

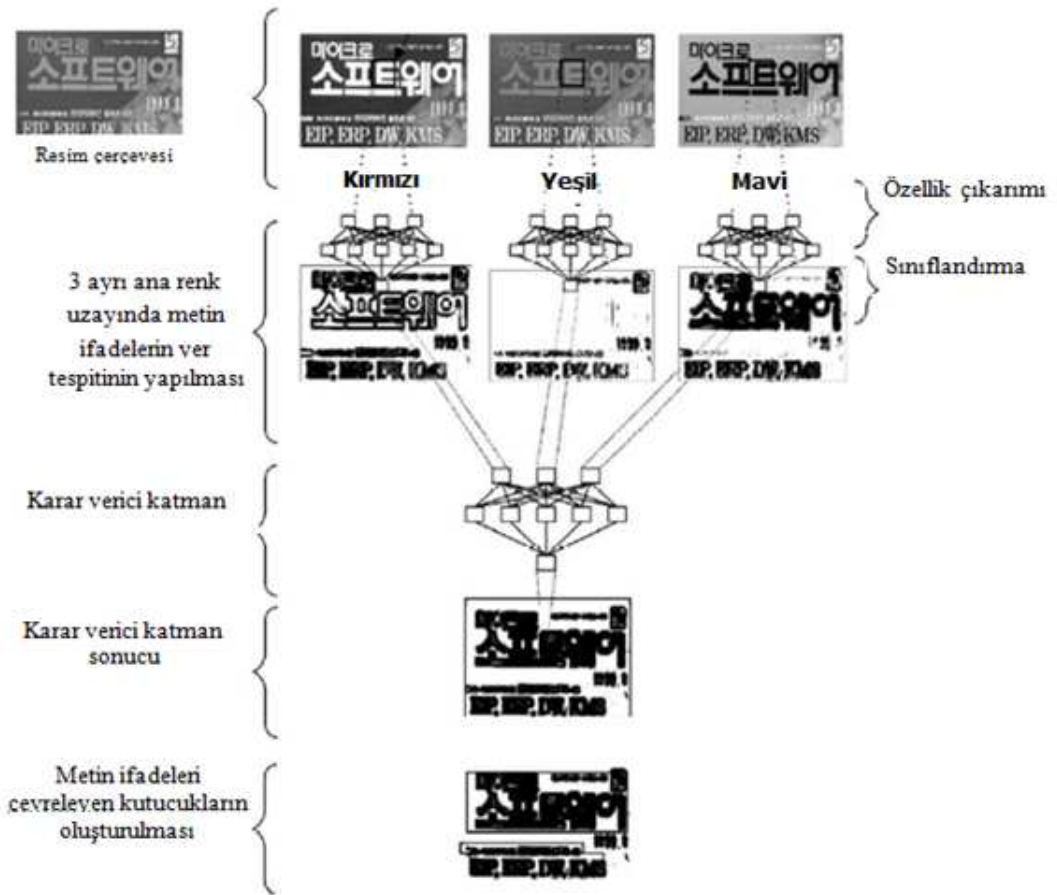
- Ana hat yaratımı: Ana hatlar bölümün içersinde yer alan önemli köşeleri tespit ederek yaratılmaktadır.
- Ana hat filtreleme: Yatay bir metin ifadeye ait olmadığı düşünülen ana hatlar elenmektedir.
- Ana hat kümeleme: Aynı metin ifadeye ait olduğu düşünülen ana hatlar birleştirilmektedir.
- Çip filtreleme: Metin ifade içermediği düşünülen çipler elenmektedir.
- Çip genişletme: Aynı metin ifadeye ait olduğu düşünülen çipler kümelenecek birleştirilmektedir.

Çip üretimi sonucunda, doku bölümlene tekniği kullanılarak, metin ifade içermesi muhtemel resim parçalarının yerleri belirlenmektedir. Belirlenmiş bu parçalar üzerinde histogram tabanlı bir algoritma ile metin ifadeler ile arka planı ayırabilecek eşik değeri belirlenmektedir.

Resim içersinden metin ifadelerin elde edilmesi için yapılan çalışmaların bir kısmı, öğrenme tabanlı yöntemleri içine alarak, metin ifadeleri tespit edebilen filtreler elde etmeyi amaçlamaktadır. Jain ve Zhong tarafından 1996'da geliştirilen algoritma

resim çerçevesini parçalara bölmekte ve her bir parçayı metin ifade, resim ve yarı ton resim olmak üzere üç farklı kategoriden birine sokmaktadır.

Jung tarafından 2001 yılında yapay sinir ağları kullanılarak renkli resimlerde metin ifadelerin yerini tespit etmek için bir çalışma yapılmıştır. Bu yöntemde RGB uzayında yer alan üç ana renk olan kırmızı, yeşil ve mavi için tüm aşamalar ayrı ayrı yapılmaktadır ve 4 katmanlı bir yapay sinir ağı kullanılmaktadır.



Şekil 6.2. Sınıflandırma yapan yapay sinir ağının yapısı

Yöntemde iki farklı yapay sinir ağı yapısı kullanılmaktadır. Bunlardan ilki RGB renk uzayındaki üç renk bandı için ayrı ayrı çalıştırılmakta ve girdi olarak sunulan resmi oluşturan parçalar üzerinde işlem yapmaktadır. Çok katmanlı ileri beslemeli yapay sinir

ağı sınıflandırıcısı parametre olarak bir pikselin renk değerini ve bu piksele komşu olan piksellerin renk değerlerini almaktadır. Çıkış düğümünde yer alan aktivasyon değerlerine bakılarak kontrol edilen piksel, metin ya da metin olmayan sınıftan birine atanmaktadır. Karar verici katman da ise, bir önceki katmanda yer alan üç farklı yapay sinir ağı sonucu değerlendirilerek son karar verilmektedir.

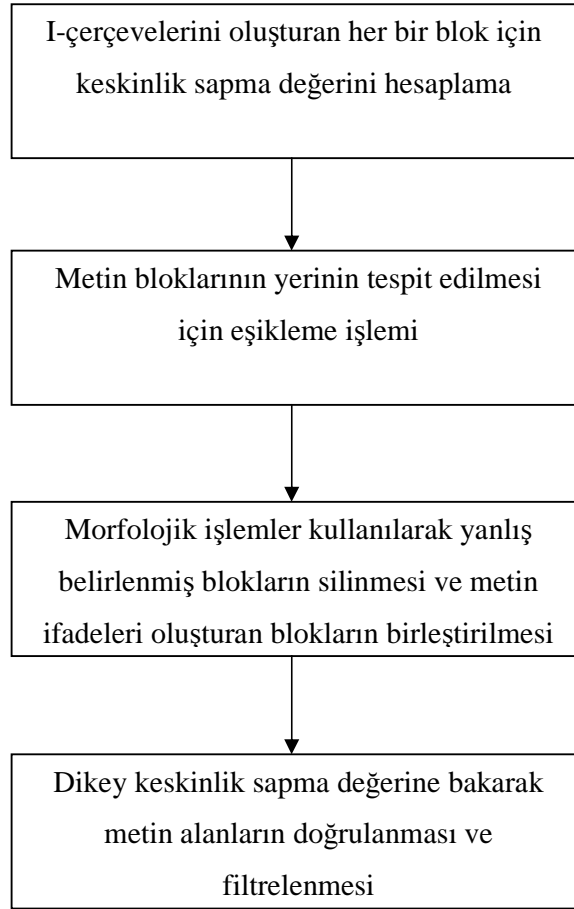
Kim v.d. (Kim v.d., 2001) tarafından destek vektör makineleri sınıflandırıcısı kullanılarak hareketli video görüntüleri içersinden metin ifadelerin yerini belirlemek için kullanılan bir yöntem geliştirilmiştir. Destek vektör makineleri, örüntü tanıma işlemlerinde oldukça başarılı sonuçlar elde edebilen sınıflandırıcılar olarak tanımlanmaktadır. Kim tarafından geliştirilen yöntemde her piksel, komşu piksellerin doku özelliklerine bakılarak metin ya da metin olmayan olmak üzere iki sınıftan birine atanmaktadır. Geliştirilen yöntemde sınıflandırma yapılırken, doku özelliği olarak açıkça bir yapı belirlenmemiş olup, öğrenme kabiliyetine sahip destek vektör makinelerinin sadece piksellerin gri tonlu renk değerlerini kullanarak bu ayrımı yapabileceği öngörülmektedir.

Tekinalp 2002 yılında video içindeki yazıları bulan, okuyan ve büyük bir video veritabanlarının endekslenmesi ve yönetilmesine olanak sağlayan bir tez çalışması yapmıştır. Bu çalışmada metin ifadelerin doku, yüksek karışıklık ve düzgün renge sahip olma özelliği dikkate alınarak video içersinde metin ifadeleri elde edebilen yeni bir yöntem ortaya konulmuştur.

## **6.2. Sıkıştırılmış Dosya İçersinden Metin İfadeleri Tespit Eden Yöntemler**

Video OCR sistemleri çoğu zaman sıkıştırılmış dosya yapılarını desteklemektedir. Sıkıştırılmış dosyanın tamamının açılmasına gerek kalmadan, video görüntüsü oluşturan resim çerçeveleri içersinde yer alan metin ifadeleri tespit edebilen yöntemler geliştirilmiştir. Bu yöntemler, metin ifadeleri tespit ederken ayırık kosinüs dönüşümü sonucunda elde edilmiş olan matrise bakarak ya da hareket vektörlerini inceleyerek öngörülerde bulunmaktadır.

Zhong v.d. tarafından geliştirilen yöntemde (Zhong v.d., 2000), sıkıştırılmış video dosyasının tamamı açılmadan, sadece I-çerçevelerinin çözülmesi sırasında kullanılan ayırık kosinüs dönüşümü katsayılarına bakılarak yazıların doku özellikleri tespit edilmeye çalışılmaktadır. Bu yöntem, dosyanın tamamı düşünüldüğünde çok ufak bir kısmı üzerinde işlem yapmaktadır.



Şekil 6.3. Sıkıştırılmış video dosyası içerisinde yer alan metin ifadelerin yerini tespit eden algoritma

Şekil 6.3'de Zhong v.d. tarafından oluşturulmuş olan algoritma görülmektedir. Algoritmanın ilk aşamasında, sıkıştırılmış dosya içerisinde yer alan ayırık kosinüs dönüşümü katsayılarının yatay metin enerji değerleri hesaplanmaktadır. Yatay metin enerji değerleri hesaplanırken kaç adet katsayının kullanılacağı yöntem parametre



olarak gönderilmektedir. Hesaplanan bu değerler üzerinde eşikleme işlemi yapılarak yüksek yatay metin enerjisine sahip bloklar belirlenmektedir. Bu bloklar, metin ifade içeren blok olmaya aday kabul edilmektedir. Aday kabul edilen bloklar arasında, içersinde metin ifade içermeyen bloklar da bulunmaktadır. Bu bloklar arasından yatay çizgi üzerinde konumlanmayanlar elenmektedir. Yöntemin ikinci aşamasında ise, metin ifade içeren blokların dikey keskinlik sapmasının da komşu bloklara göre farklılık göstermesi gerekliliği dikkate alınmaktadır. Yöntemin bu aşamasında blokların dikey metin enerjileri hesaplanmaktadır. Dikey metin enerjisi yüksek olmayan blokların metin ifade içermediği kabul edilmektedir.

Gu tarafından sunulan bir yöntem (Gu, 2001) de Zhong v.d. tarafından sunulan yönteme benzerlik göstermektedir. Zhong v.d. tarafından önerilen yöntemde metin ifade içeren blokların tespiti için ayırık kosinüs dönüşümü katsayılarının tamamı üzerinde işlem yapılmamaktadır. Yönteme parametre olarak sunulan iki değere bakılarak metin enerji değerleri hesaplanırken katsayılar dizisinin hangi elemanlarının kullanılacağı belirlenmektedir. Gu tarafından önerilen yöntem de ise, ayırık kosinüs dönüşümü katsayılarından DC katsayısı hariç tüm AC katsayıların ortalaması keskin kenarlara sahip blokları belirlemek için kullanılmaktadır.

$$E_{ac}(i, j) = \left( \sum_{u=0}^7 \sum_{v=0}^7 |C_{uv}(i, j)| \right) / 63 \quad (u, v) \neq (0, 0) \quad (6-1)$$

Gu tarafından önerilen yöntemde blokların metin enerji değerleri hesaplanırken, denklem (6-1) kullanılmaktadır. I-çerçevesini oluşturan her bir parlaklık bilgisi içeren blok için metin enerji değerleri hesaplanmaktadır. Hesaplanan bu değerler kullanılarak, keskin kenarlara sahip blokları tespit edebilmek için kullanılacak bir eşik değeri elde edilmektedir. Eşik değerinin üstünde metin enerjisine sahip aday bloklar için bir kontrol daha yapılmaktadır. Metin ifade içeren blokların bir sonraki P-çerçevesinde kodlanmamış olması ya da bu bloğa ait bir hareket vektörünün kodlanmamış olması gerekmektedir. Aksi halde bu aday blok elenmektedir ve metin ifade içermediği kabul edilmektedir.

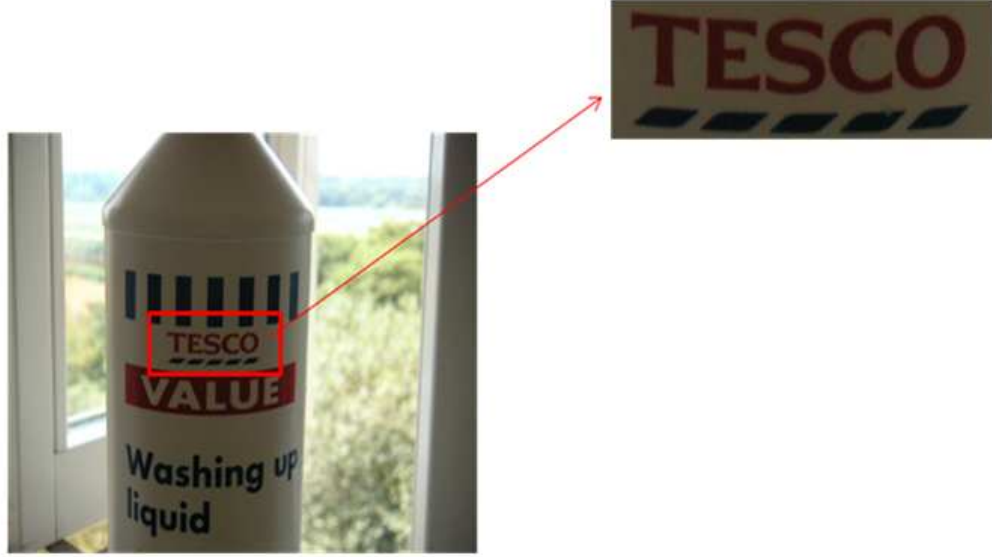
## 7. VIDEO OCR SİSTEMİNDE METİN İFADELERİ TESPİT ETMEK İÇİN ÖNERİLEN YÖNTEM

Resim içersinde yer alan metin ifadelerin konumlarını belirlemek ve metin ifadeleri tespit etmek için klasik OCR yöntemlerinin yetersiz kalmasından dolayı, doğal resimler içersinde yer alan metin ifadeleri elde etmek için birçok araştırma yapılmaktadır. Video arşivleri üzerinde sorgulama yapmaya imkân sağlayan Video OCR Sistemlerinin de en önemli bileşenleri arasında, resim içersinde yer alan metin ifadeleri tespit etmek için kullanılan bileşen sayılabilir. Bu bileşenin metin ifadeleri elde etmedeki başarısı, sistemin başarı oranını önemli ölçüde belirlemektedir.

Geliştirilen Video OCR Sistemi, MPEG 2 yöntemi kullanılarak sıkıştırılmış hareketli video görüntüleri içersinde yer alan I-çerçevelerini kullanarak içerik tabanlı sorgulama yapabilmektedir. Resim çerçeveleri içersinde yer alan metin ifadeleri elde etmek için resim çerçevesinin tamamı çözümlenmekte ve metin ifadenin yerinin tespiti bu çerçeve üzerinde yapılmaktadır.

Doğal resim içersinden metin ifadeleri elde etmek için geliştirilen yöntemde metin ifadelerin konumlarını belirlemede, metin ifadelerin karakteristik özellikleri referans oluşturmaktadır. Bu karakteristik özellikler;

- Karakterler arka plan ile zıtlık içermektedir.
- Karakterler genellikle tek renklidir.
- Karakterlerin belirli bir boyut sınırlaması vardır.
- Karakterler genellikle yatay olarak aynı hizada olmak zorundadır.
- Karakterler arasında belirli bir aralık bulunmak zorundadır.



Şekil 7.1. Örnek resim

Şekil 7.1’de örnek resim görülmektedir. Bu resim içerisinde yer alan TESCO yazısı kırmızı olarak işaretlenmiş olup, aşağıda listelenmiş olan karakteristik özelliklere sahiptir.

- TESCO metni arka plan ile zıt renklere sahiptir.
- Metni oluşturan karakterler tek renklidir. (kırmızı)
- Metni oluşturan karakterler aynı boyuttadır.
- Metni oluşturan karakterler yatay olarak aynı hizadadır.
- Metni oluşturan karakterler arasında belirli bir mesafe bulunmaktadır.

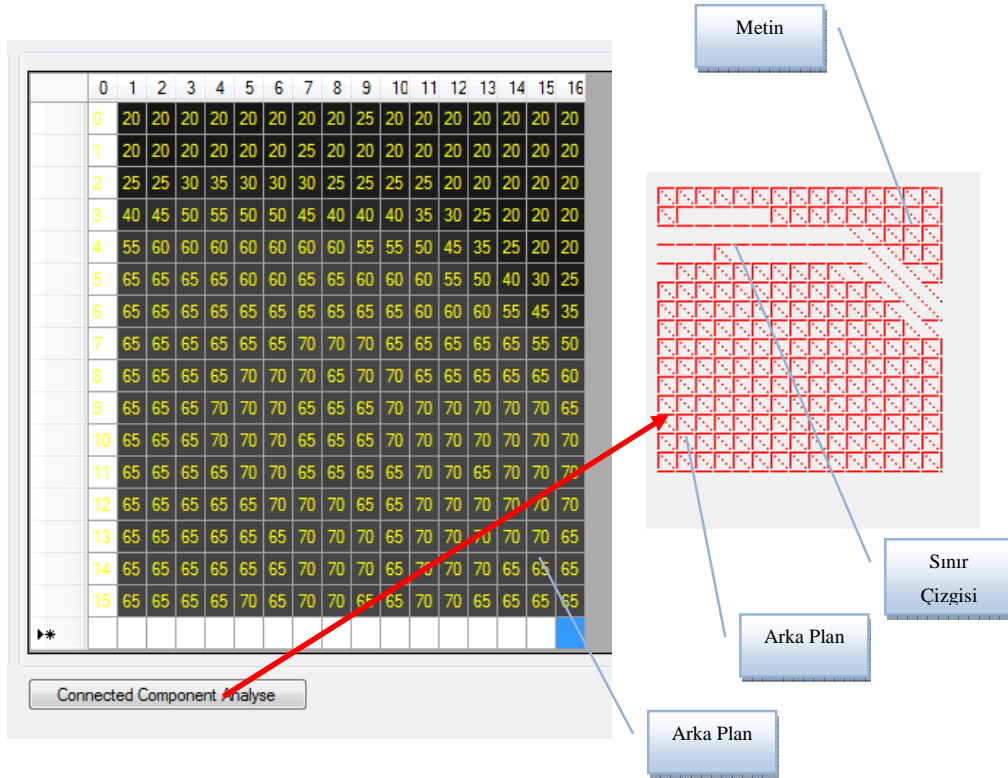
Doğal resim içerisinde yer alan metin ifadelerin birbirleri ile ortak renk değerleri içermesi, bağlı bileşen analizinin yapılmasını mümkün kılmaktadır.



Şekil 7.2. Örnek resim içerisinde yer alan bir karakter

Şekil 7.2’de örnek resim içersinde yer alan bir karakter oldukça yakınlaştırılmış halde görülmektedir. Dikkat edilirse, karakterler ile arka plan arasındaki ayrımın, sadece renk değerleri arasındaki farklılık kullanılarak sağlanabileceği görülmektedir. Bu işlem için bağlı bileşen analizini yapmak yeterli olmaktadır.

Şekil 7.3’de örnek resmin renk değerlerini 16 \*16 bloklar olarak gösteren basit bir örnek program görülmektedir. Renk değerleri gri tonlu olarak gösterilmekte olup, karakteri oluşturan renk değerleri kendi aralarında birbirine yakın, arka planı oluşturan renk değerleri de kendi aralarında birbirine yakın değerlere sahiptir.



Şekil 7.3. Örnek Uygulama

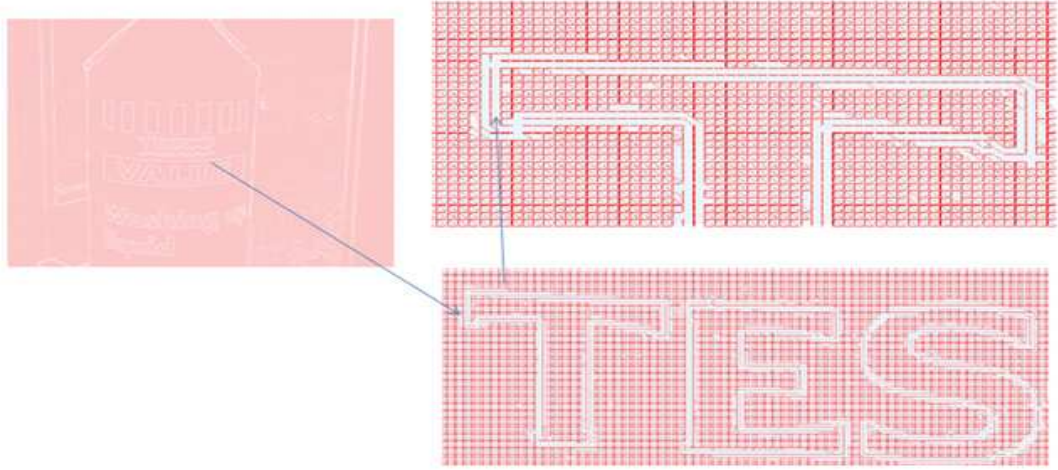
Şekil 7.3’de yer alan uygulama bağlı bileşenler analizini de yapmaktadır. Yazı ile arka plan arasında, sınır çizgisi olarak görülen bir bölüm yer almaktadır. Bu sınır çizgisi, renk değeri olarak arka plan ve metin renk değerleri arasında renk değerlerine sahiptir. Arka plan 60, 65 ve 70, metin 20, 25 ve 30 sınır çizgisi ise 40, 45 ve 50 renk

değerlerine sahiptir. Renk değerlerinin sürekli 5 ve 5 in katlarında olduğu görülmektedir. Bunun nedeni arka plan, sınır çizgisi ve metin arasındaki sınırı daha keskin belirlemek için normalizasyon işleminin yapılmış olmasıdır. Şekil 7.4’de normalizasyon işlemini yapan fonksiyon görülmektedir.

```
private int normalize2(int value, int divisor)
{
    return (value - (value - (value / divisor) * divisor));
}
```

Şekil 7.4. Normalizasyon işlemi yapan fonksiyon

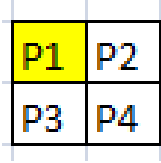
Resmin tamamı üzerinde, bağlı bileşen analizi fonksiyonu çalıştırıldığında Şekil 7.5’de yer alan sonuç elde edilmektedir.



Şekil 7.5. Bağlı bileşenler analizinin tüm resim üzerinde uygulanması

### 7.1. Bağlı Bileşen Analizi

Bağlı Bileşen Analizi, resmi oluşturan piksellerin her birinin kendi renk değeri ile komşu piksellerinin renk değerlerini karşılaştırmaktadır. Şekil 7.6’da p1 referans pikseli, p2, p3 ve p4 ise bu pikselin komşu piksellerini göstermektedir.



Şekil 7.6. Referans Piksel ve Komşu Pikseller

Bir pikselin arka plana ya da metin ifadeye ait olup olmadığına, komşu piksellerinin renk değerleri ile olan benzerliğine bakılarak karar verilmektedir. Bir pikselin komşu piksel ile renk değeri arasındaki mutlak fark 5 olduğunda, bu piksel ile komşu piksel birbirine bağlıdır denilmektedir.

Metin ifadeyi tespit etmek için ortaya konulan yeni yöntem için **bir pikselin bağlı komşu piksel sayısı 3'ten düşük ise, bu piksel sınır çizgisine ait bir pikseldir** ifadesi kuraldır. Bu kural doğrultusunda, resmi oluşturan tüm piksellerin sınır çizgisine arka plana ya da metin ifadeye mi ait olduğu tespit edilmektedir. Şekil 7.7'da bir referans piksele bağlı pikselleri elde etmek için kullanılan **checkPixel()** fonksiyonu görülmektedir.

```

public void checkPixel(int i,int j)
{
    if (j < b.Height-1)
    {
        if (Math.Abs(pixelValues[i, j].grayValue - pixelValues[i, j + 1].grayValue) <= 5)
        {
            pixelValues[i, j].connectedPixelPositions.Add(new position(i, j + 1));
        }
    }
    if (i < b.Width -1)
    {
        if (Math.Abs(pixelValues[i, j].grayValue - pixelValues[i + 1, j].grayValue) <= 5)
        pixelValues[i, j].connectedPixelPositions.Add(new position(i + 1, j));
    }
    if (i < b.Width-1 & j <b.Height-1)
    {
        if (Math.Abs(pixelValues[i, j].grayValue - pixelValues[i + 1, j + 1].grayValue )
        <= 5)
        {
            pixelValues [i, j].connectedPixelPositions.Add(new position(i+1, j + 1));
        }
    }
}

```

Şekil 7.7. checkPixel() fonksiyonu

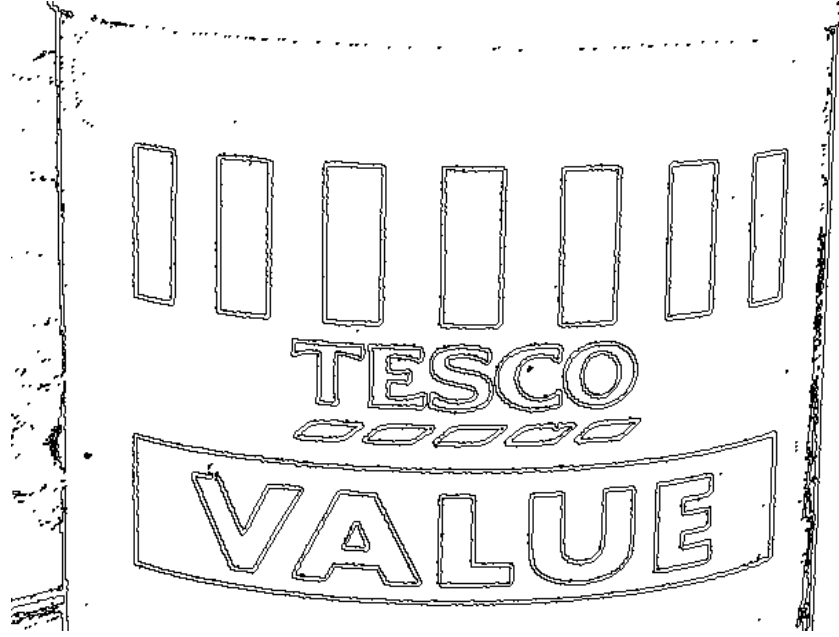
Resim üzerinde komşu pikselleri tespit eden fonksiyon çalıştırdıktan sonra, sadece komşu piksel sayısı 3'ten az olanları siyah renk olarak atayarak görüntülediğinde Şekil 7.8'de yer alan sonuç resim elde edilmektedir. Bu sonuç resim, kuralda belirtildiği üzere resim içerisinde yer alan nesnelerin sadece sınır çizgilerini içermektedir.



Şekil 7.8. Resim içerisinde yer alan nesnelerin sınır çizgilerinin belirlenmesi

## 7.2. Resmin Sınır Çizgilerinden Metin İfadeleri Elde Etme

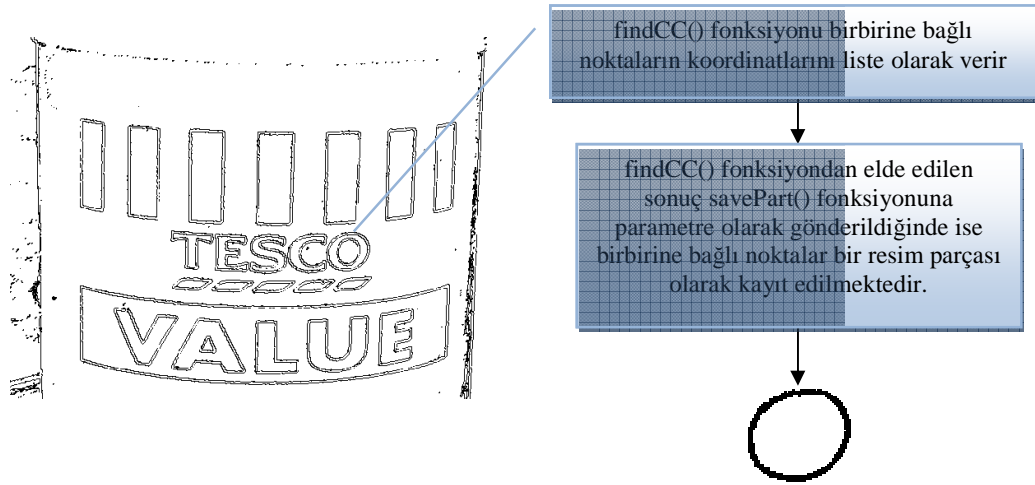
Şekil 7.8'de yer alan sonuç resim nesnelerin sınır çizgilerini içermektedir. Şekil 7.9'da bu resmin daha yakınlaştırılmış hali görülmektedir. Dikkat edilirse, metin ifadelerin ve resim içerisinde yer alan objeleri çevreleyen sınır çizgilerinin birbirine bağlı olduğu görülmektedir.



Şekil 7.9. Sonuç resmin yakınlaştırılmış hali

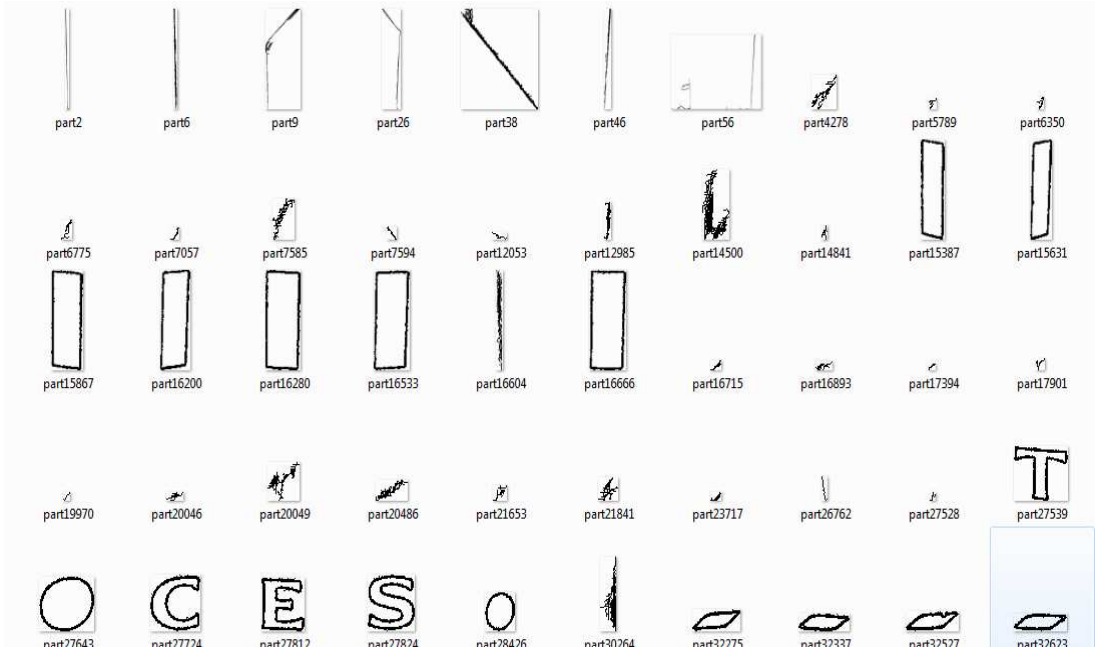
Yöntemin bundan sonraki adımında, birbirine bağlı olan pikseller tespit edilmektedir. Yöntem içerisinde, birbirine bağlı olan pikselleri tespit etmek için **finCC()** adlı bir fonksiyon kullanılmaktadır. Bu fonksiyon çalıştırıldığında  $x$ ,  $y$  konumundaki piksele bağlı olan piksellerin koordinatları bir liste halinde elde edilmektedir. Elde edilen bu bağlı koordinatların listesi yöntem içerisinde tanımlanan **savePart()** fonksiyonu kullanılarak bir resim parçası olarak kayıt edilmektedir. Şekil 7.10'da **findCC()** ve **savePart()** fonksiyonlarının çalıştırılması sonucunda elde edilen sonuç görülmektedir. **findCC()** fonksiyonu sınır çizgisi olan bir resim parçası üzerinde işletildiğinde o sınır çizgisinin üzerinde yer alan tüm piksellerin listesini geri değer olarak döndürmektedir. **savePart()** fonksiyonu da bu değerler listesini kullanarak o sınır çizgisine ait karakter ya da resim parçasını bütünüyle elde etmektedir.





Şekil 7.10. findCC ve savePart fonksiyonlarının kullanımı

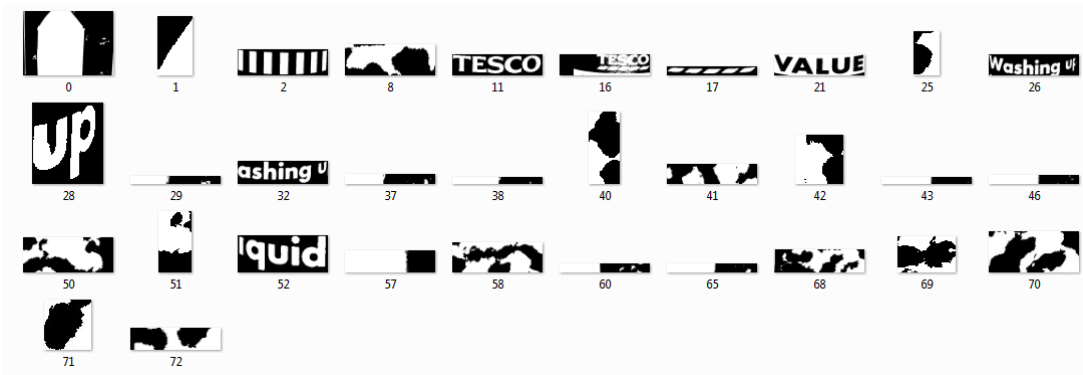
Şekil 7.11’de örnek resim için tüm bağlı piksellerin **savePart()** fonksiyonu ile kayıt edilmiş hali yer almaktadır. Görüldüğü üzere bu parçaların birçoğu metin ifadeyi oluşturan karakterler değildir. Metin ifadeyi oluşturan parçaların dışında kalan parçaların OCR işlemine tabi tutulmayarak yok sayılması gerekmektedir. Ayrıca OCR işleminin düzgün netice verebilmesi için bu parçaların iç kısımlarının siyah renk ile doldurulması gerekmektedir.



Şekil 7.11. Resim parçaları

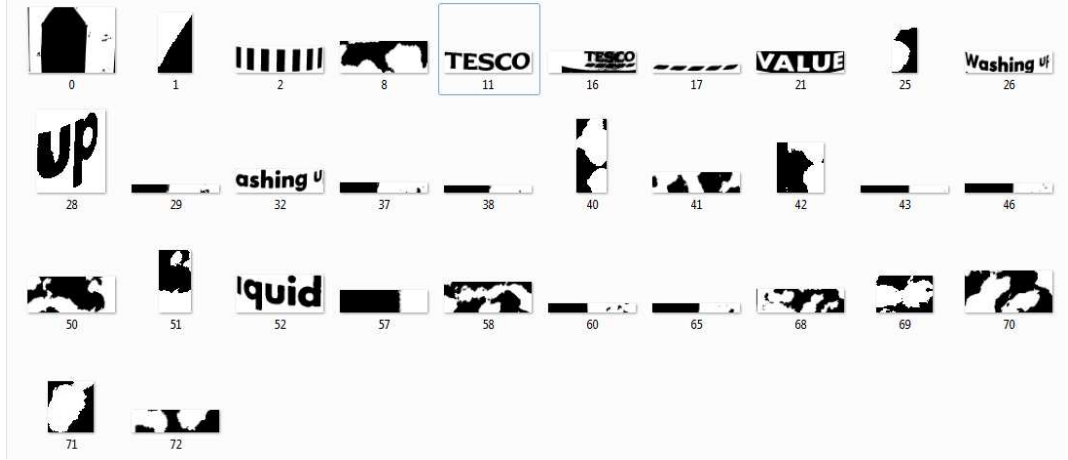
### 7.3. Metin İfade Olabilecek Parçaların Birleştirilmesi

Resim parçalara ayrıldıktan sonra, OCR işleme tabi tutulmaktadır. Ancak bu işlemden önce, yükseklik olarak birbirine yakın değerler içeren, konum olarak da birbirine yakın olan parçaların birleştirilmesi, OCR işleminin daha verimli olmasını ve daha anlamlı metin ifadeler üretmesini sağlamaktadır.. Yöntem içerisinde yer alan resim parçalarından ortak konum ve yükseklikte olanlar **commonParts()** fonksiyonu ile birleştirilmektedir.



Şekil 7.12. Birleştirilmiş Resim parçaları

Şekil 7.12’de **commonParts()** fonksiyonun çalıştırılmasından sonra elde edilen resim parçaları görülmektedir. Yapılan denemelerde, OCR işleminin düzgün netice verebilmesi için, resim parçalarının ters dönüştürülmüş olanlarının da işlenmesi gerekmektedir. Örneğin OCR işlemi Şekil 7.12’de yer alan VALUE kelimesini düzgün elde edememektedir. Buna karşın Şekil 7.13’de yer alan bu resimlerin ters renk değerlerine sahip olanları üzerinde işlem yaptığında ise, tanıma işlemi gerçekleşmektedir. Bu yüzden OCR işleminin sonuç parçalar ve bu parçaların ters renk değerlerine sahip olanları üzerinde yapılması daha etkin sonuç vermektedir.



Şekil 7.13. Birleştirilmiş Resim parçalarının ters dönüştürülmüş halleri

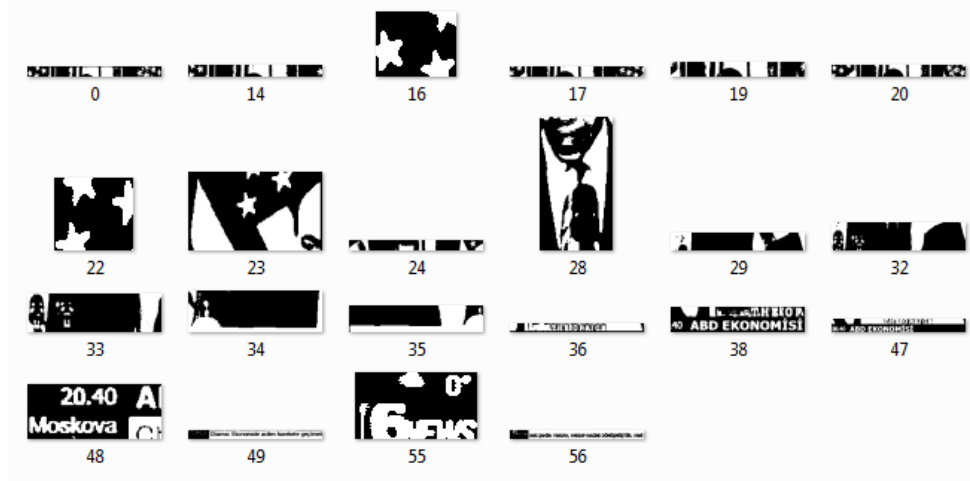
## 8. GELİŞTİRİLEN VIDEO OCR SİSTEMİ

Bu tez çalışmasında geliştirilen Video OCR sistemi, MPEG-2 yöntemi kullanılarak sıkıştırılmış olan video görüntüsü içeren dosya içerisinde yer alan I-çerçevelerini elde ederek yedinci bölümde ortaya konulan yöntemi kullanarak bu çerçeveler içerisinde yer alan metin ifadelerin yerlerini tespit etmektedir. Yerleri tespit edilen bu metin ifadeler, hazır bir OCR bileşeni tarafından işlenmektedir.



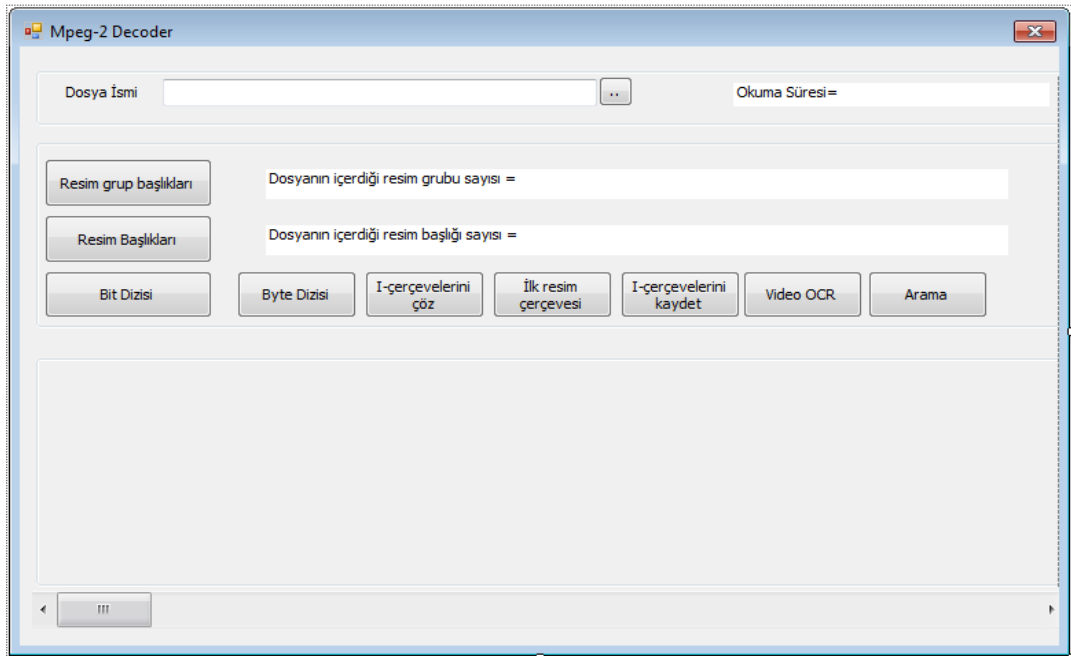
Şekil 8.1. Örnek video dosyası içerisinde elde edilmiş I-çerçevesi

Şekil 8.1’de örnek video dosyası içersinden elde edilmiş olan I-çerçevesi görülmektedir. Bu çerçeve üzerinde yedinci bölümde ortaya konulan yöntem kullanıldığında Şekil 8.2’de yer alan resim parçaları ile bu resim parçalarının ters renk değerlerine sahip olanları elde edilmektedir. Önerilen yöntem, doğal bir resim içerisinde metin ifadeler içeren alanları tespit etmektedir. Resim çerçevesinin tamamı OCR işlemine tabi tutulduğunda hiçbir başarılı sonuç alınamazken, önerilen yöntem tarafından elde edilen resim parçaları OCR işlemine tabi tutulduğunda başarılı sonuçlar elde edilmektedir.



Şekil 8.2. Örnek resim çerçevesi üzerinde önerilen yöntemin uygulanması sonucunda elde edilen resim parçaları

Geliştirilen Video OCR Sisteminin ara yüzü Şekil 8.3'te görülmektedir. Dosya seçimi yapıldıktan sonra, ilk olarak **Resim Başlıkları** tuşuna basılarak, tüm dosya içerisinde yer alan resim başlıklarının konumları belirlenmektedir.



Şekil 8.3. Geliştirilen Video OCR Sisteminin ara yüzü

**I-çerçeveslerini kaydet** tuşuna basıldığında, sıkıştırılmış video görüntüsünü oluşturan I-tipi çerçeveler bir klasör içerisinde kayıt edilmektedir. Kayıt edilen I-çerçevelerine atanacak olan isim, bu çerçevenin video içerisinde kaçınca saat, dakika ve saniye de görüntülediği bilgisi ile birlikte, resim başlığından elde edilen geçici referans numarasını içermektedir.



Şekil 8.4. Örnek video görüntüsü üzerinden elde edilmiş olan I-çerçeveslerinin bir bölümü


Şekil 8.4’de geliştirilen Video OCR sistemi tarafından **I-çerçeveslerini kaydet** tuşuna basılarak elde edilen resim çerçevelerinin bir bölümü görülmektedir. İlk resim çerçevesi 0. Saat, 0. Dakika, 1. Saniye’de gösterilmektedir ve geçici referans numarası olarak 2 değerine sahiptir.

**Video OCR** tuşuna basıldığında ise, kayıt edilmiş bu klasör içerisinde yer alan dosyalar üzerinde metin tanıma işlemi yapılarak, resim içersinden tespit edilen karakter ifadeler veritabanına kayıt edilmektedir. **Arama** tuşuna basılarak da, anahtar bir kelime belirlenerek, bu kelimenin video içerisinde yer aldığı bölümden başlayarak video görüntülenmektedir. Şekil 8.5’de arama işlemi için kullanılan ara yüz yer almaktadır. Anahtar kelimeyi içeren, resim çerçevelerinin konumları listelenmiştir. Kullanıcı bu listede yer alan herhangi bir satıra tıkladığında, o resim çerçevesini içeren video dosyası medya oynatıcısına yüklenerek, video o resim çerçevesinin video içerisinde yer aldığı noktaya konumlanarak oynatılmaktadır.

Veritabanı içerisinde arama

Anahtar Kelime: OBAMA

Dosya ID	Dosya Adı	Anahtar Kelime	Saat	Dakika	Saniye
1	c://cem/b.mpg	1Obama:	0	1	16
1	c://cem/b.mpg	--*~20.39Edime...	0	1	17
1	c://cem/b.mpg	Obama:	0	1	18
1	c://cem/b.mpg	***;ONOMISIO...	0	1	19
1	c://cem/b.mpg	" B ...N+Obama:	0	1	20
1	c://cem/b.mpg	1Obama:	0	1	21
1	c://cem/b.mpg	3'Obama:	0	1	22
1	c://cem/b.mpg	3*GevsiNOMIO...	0	1	23
1	c://cem/b.mpg	ABAŞab+ONVO...	0	1	24
1	c://cem/b.mpg	ABOObama:	0	1	25
1	c://cem/b.mpg	4*Obama:	0	1	26
1	c://cem/b.mpg	Obama:	0	1	27



20:39 ABD EKONOMİSİ  
Edime  
6 NEWS  
Obama: Ekonomide acilen harekete geçilmeli  
S TARAFINDAN DİNLENMEYE BAŞLANMIŞTIR SÜREÇTE YAPILMA  
Playing b' 01:18

Şekil 8.5. Arama işlemi için kullanılan ara yüz

## 9. SONUÇ

Bu tez çalışmasında, MPEG-2 video sıkıştırma yöntemi kullanılarak sayısal ortamda kayıt edilmiş dosyalar içerisinde yer alan metin ifadeleri tespit ederek sorgulamaya imkân sağlayan bir Video OCR Sistemi geliştirilmektedir.

Video OCR Sisteminin performansını önemli ölçüde etkileyen, resim çerçeveleri içerisinde yer alan sahne metni ve yapay metinlerin konumlarını tespit etmek için yeni bir yöntem ortaya konulmaktadır. Resim çerçevesinin tamamı üzerine OCR işlemi uygulandığında, metin ifadelerin elde edilmesinde başarılı sonuçlar alınmazken, tezde gerçekleştirilen yöntem kullanılarak resim çevreleri içerisinde yer alan metin ifadelerin konumları tespit edilerek OCR işlemi uygulandığında başarılı sonuçlar elde edilmektedir. Elde edilen sonuçlar, veritabanına kayıt edilerek, hareketli video görüntüsünün içerik tabanlı sorgulanması mümkün olmaktadır.

Bu tez çalışmasında geliştirilen Video OCR Sistemi MPEG-2 yöntemini desteklemektedir. MPEG-2 sıkıştırma yönteminde, tek bir dosya içerisinde aynı anda birden fazla video görüntüsü ve ses bulunabilmektedir. Bu tez çalışmasında, MPEG-2 sıkıştırma yönteminde kullanılan yapılar en alt seviyeye kadar çözümlenmiş olup, bundan sonraki yapılacak olan çalışmalarda referans olarak kullanılacak bilgiler elde edilmiştir. (Ses analiz, MPEG-2 video şifreleme işlemleri...)

Bu çalışma da MPEG-2 dosya yapısı tamamen çözümlendiğinden, MPEG-2 yönteminin diğer versiyonları olan MPEG-1, MPEG-4, MPEG-7, H.264... yöntemlerini çözümlmek ve bu yöntemler içerisinde yer alan yapıları anlamak daha basit olacaktır.



**KAYNAKLAR**

Bukhari S. S., Shafait F., Breuel T. M., 2009, "Coupled Snakelet Model for Curled Textline Segmentation of Camera-Captured Document Images", 10th International Conference on Document Analysis and Recognition, 61-65

Dataong C., Jean-Marc O., Herve B., 2004, "Text detection and recognition in images and video frames", The Journal Of The Pattern Recognition Society, 37, 595-608

Ezaki N., Bulacu M., Schomaker L., 2004, "Text Detection from Natural Scene Images: Towards a System for Visually Impaired Persons", Proceedings of 17th International Conference on Pattern Recognition, 2, 683-686

Ferreire S., Garin V., Gosselin B., 2005, "A Text Detection Technique Applied in the Framework of a Mobile Camera-Based Application", 5th International Workshop on Camrea-Based Document Analysis and Recognition

Gargi U., Crandall D., Antani S., Gandhi T., Keener R., Kasturi R., 1999, "A System for Automatic Text Detection in Video", Proceedings of the Fifth International Conference on Document Analysis and Recognition, 29-32

Gllavata J., Ewerth R., Freisleben B., 2003, "A Robust Algorithm for Text Detection in Images", Proceedings of 3rd International Symposium on Image and Signal Processing and Analysis, 611-616

Gu L., 2001, "Text Detection and Extraction in MPEG Video Sequences", CBMI '01, Brescia, Italy, September 19-21

International Standard ISO/IEC 13818-2, 1995

Iwamura M., Tsuji T., Horimatsu A., Kise K., 2009, "Real-Time Camera-Based Recognition of Character and Pictograms", 10th International Conference on Document Analysis and Recognition, 76-80

Jung K., 2001, "Neural network-based text location in color image", Pattern Recognition Letters, 22, 1503-1515

Kim E., Lee S., Kim J., 2009, "Scene Text Extraction using Focus of Mobile Camera", 10th International Conference on Document Analysis and Recognition, 166-170

Kim K., Jung K., Park S. H., Kim H. J., 2001, "Support vector machine-based text detection in digital video", The Journal Of The Pattern Recognition Society, 34, 527-529

Lan Z., Zhao J., 2008, "A Method of News Caption Location Based on C-Mean and Edge Detection, Congress on Image and Signal Processing", 620-624

Lee C. M. , Kankanhalli A., 1995, "Automatic Extraction Of Characters In Complex Scene Images", International Journal Of Pattern Recognition And Artificial Intelligence, 9(1), 67-82

Lim Y. K., Choi S.H., Lee S.W., 2000, "Text Extraction in MPEG Compressed Video for Content-based Indexing", 15th International Conference on Pattern Recognition, Barcelona, Spain, September 3-8

Liu Q., Jung C., Kim S., Moon Y., Kim J.Y., 2006, "Stroke Filter for Text Localization in Video Images", International Conference on Image Processing, Atlanta, GA, USA, October 8-11

Liu X., Samarabandu J., 2006, "Multiscale Edge-Based Text Extraction From Complex Images", IEEE International Conference on Multimedia and Expo, Totonto, Canada, July 09-12

Lui Z., Sarkar S., 2008, "Robust Outdoor Text Detection Using Text Intensity and Shape Features", 19th International Conference On In Pattern Recognition, 1-4

Lyu M. R., Song J., Cai M., 2005, "A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction", IEEE Transactions On Circuits And Systems For Video Technology, 15(2), 243-255

Ngo C., Chan C., 2005, "Video text detection and segmentation for optical character recognition", Multimedia Systems 10, 261-272

Pan Y.F., Hou X., Lui C. L., 2009, "Text Localization in Natural Scene Images based on Conditional Random Field", 10th International Conference on Document Analysis and Recognition, 6-10

Quixiang Y., Jianbin J., Jun H., Hua Y., 2007, "Text detection and restoration in natural scene imaes", Journal Of Visual Communication And Image Representation", 18, 54-513

Sato T., Kanade T., Hughes E. K., Smith M. A., Satoh S., 1999, "Video OCR: indexing digital new libraries by recognition of superimposed captions", Special Section on video labraries, 385-395

Second Edition of International Standard ISO/IEC 13818-1, 2000

Smith M.A., Kanade T., 1995, "Video skimming for quick browsing based on audio and image characterization", Technical Report CMU-CS, 186

- Tekinalp S., 2002, "Detecting and Recognizing Text From Video Frames", Doktora tezi, Orta Doğu Teknik Üniversitesi
- Trier Q., Jain A. K., 1995, "Goal-Directed Evaluation of Binarization Methods", IEEE Transactions On Pattern Analysis And Machine Intelligence, 17(12)
- Vonikakis V., Andreadis J., Papamarkos N., 1997, "Adaptive Document Binarization", Proceedings of the 4th International Conference on Document Analysis and Recognition, 147-152
- Wang W., Gao W., Li J., Lin S., 2000, "News Content Highlighting via Fast Caption Text Detection on Compressed Video", Proceedings of the Second International Conference on Intelligent Data Engineering and Automated Learning, Data Mining, Financial Engineering, and Intelligent Agents, 443-448
- Wernicke A., Lienhart R., 2000, "On the Segmentation of Text in Videos", IEEE International Conference on Multimedia and Expo, New York City, NY, USA, July 30-August 2
- Wong E. K., Chen M., "2003", "A new robust algorithm for video text extraction", Pattern Recognition, 36, 1397-1406
- Wu V., Manmatha R., Riseman M., 1997, "Finding Text In Images", Proceedings of the second ACM international conference on Digital libraries, 3-12
- Xu J., Jiang X., Wang Y., 2008, "Caption Text Extraction Using DCT Feature in MPEG Compressed Video", World Congress on Computer Science and Information Engineering, 431-434
- Yao J. L., Wang Y. Q., Weng L. B., Yang Y. P., 2007, "Locating Text Based On Connected Component And SVM", Proceedings of the 2008 International Conference On Wavelet Analysis And Pattern Recognition, Beijing, China, November 2-4
- Zhang X. W., Zheng X. B., Weng Z. J., 2008, "Text Extraction Algorithm Under Background Image Using Wavelet Transforms", Proceedings of the 2008 International Conference On Wavelet Analysis And Pattern Recognition, Hong Kong, August 30-31
- Zhong Y., Karu K., Jain A. K., 1995, "Locating Text in Complex Color Images", Proceedings of the Third International Conference on Document Analysis and Recognition, 146-147
- Zhong Y., Zhang H., Jain A.K., 2000, "Automatic Caption Localization In Compressed Video", IEEE Transactions On Pattern Analysis And Machine Intelligence, 22(4), 385-392

**EKLER****EK-A. Tablo B-14**

<b>Değişken uzunlukta kod</b>	<b>İlerleme</b>	<b>Düzy</b>
10	Blok Sonu	
1 s	0	1
11 s	0	1
011 s	1	1
0100 s	0	2
0101 s	2	1
0010 1 s	0	3
0011 1 s	3	1
0011 0 s	4	1
0001 10 s	1	2
0001 11 s	5	1
0001 01 s	6	1
0001 00 s	7	1
0000 110 s	0	4
0000 100 s	2	2
0000 111 s	8	1
0000 101 s	9	1
0000 01	Kaçış Kodu	
0010 0110 s	0	5
0010 0001 s	0	6
0010 0101 s	1	3
0010 0100 s	3	2
0010 0111 s	10	1
0010 0011 s	11	1
0010 0010 s	12	1
0010 0000 s	13	1
0000 0010 10 s	0	7
0000 0011 00 s	1	4
0000 0010 11 s	2	3
0000 0011 11 s	4	2
0000 0010 01 s	5	2
0000 0011 10 s	14	1
0000 0011 01 s	15	1
0000 0010 00 s	16	1