

**YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HABER VİDEOLARININ GÖRÜNTÜ İŞLEME
YÖNTEMLERİ İLE HABERLERE BÖLÜTLENMESİ**

Rajab DAVUDOV

**FBE Bilgisayar Mühendisliği Anabilim Dalı Programında
Hazırlanan**

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yard. Doç. Dr. Elif KARSLIGİL

İSTANBUL, 2009

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	ii
ŞEKİL LİSTESİ	iv
ÇİZELGE LİSTESİ.....	v
KISALTMA LİSTESİ	vi
ÖNSÖZ.....	vii
ÖZET	viii
ABSTRACT	ix
1. GİRİŞ.....	1
1.1. Önceki Çalışmalar	2
2. HABER PROGRAMI BÖLÜTLEME SİSTEMİ TASARIMI	4
2.1. Renk ve Kenar Özelliklerinin Analizi Katmanı	5
2.1.1. Renk Değişiminin Analizi	6
2.1.2. Kenar Değişimi Analizi	6
2.1.3. Canny Kenar Belirleme Yöntemi	8
2.1.3.1. Resmin Gauss Filtresi ile Yumuşatılması.....	8
2.1.3.2. Sobel Maskeleri ile Kenar Piksellerin Değer ve Yönünün Belirlenmesi	10
2.1.3.3. En Büyük Olmayanı Bastırma (Non-Maximum Suppression).....	11
2.1.3.4. Çift Eşik Seviyesi (Hysteresis Thresholding) Uygulanması	11
2.2. Spiker Tanıma Yöntemi İle Sahne Geçişlerinin Bulunması Katmanı.....	12
2.2.1. Yüz Bulma.....	13
2.2.2. Ana Bileşen Analizi Yötemi İle Yüz Tanıma.....	13
2.2.3. Yüz Tanıma Başarısının Değerlendirilmesi	15
2.3. Yazı İçeriğinin Analizi Katmanı.....	15
2.3.1. Yazı Bölgesi Karşılaştırma İşlemi.....	16
2.3.2. Otsu Eşik Seviyesi Bulma Yöntemi	16
3. ÖN HAZIRLIK	19
3.1. Haber Videosu Veritabanının Oluşturulması	19
3.2. Renk ve Kenar Değişimi Özellikleri Katmanı İçin Ön Hazırlık	19
3.2.1. Renk Değişimi Analizi İçin Eşik Seviyesi Belirlenmesi	20
3.2.2. Kenar Değişimi Analizi İçin Eşik Seviyesi Belirlenmesi.....	23
3.3. Spiker Tanıma Katmanı İçin Ön Hazırlık.....	27
3.2.3. Yüz Tanıma Sisteminin Eğitilmesi.....	27
3.2.4. Eşik Seviyelerinin Belirlenmesi	29
3.3. Yazı Tanıma Katmanı İçin Öz Hazırlık.....	32
3.4. Sistem Aşamalarının Özeti	34
4. UYGULAMA.....	35

4.1.	Farklı Kanallar İçin Haber Programları ile Testler.....	36
4.1.1.	Kanal-1 Haber Programı İçin Değerlendirme	37
4.1.2.	Kanal-2 Haber Programı İçin Değerlendirme	38
4.1.3.	Kanal-2 Haber Programı İçin Değerlendirme	39
4.1.4.	Kanal-3 Haber Programı İçin Değerlendirme	41
4.1.5.	Kanal-3 Haber Programı İçin Değerlendirme	42
4.2.	Test Sonuçlarının Değerlendirilmesi	43
4.2.1.	Yüz Tanımadaki Hatalar.....	44
4.2.2.	Haber İçeriğinin S-V Yapısına Uymaması.....	44
4.2.3.	Haber Özetleri.....	45
4.2.4.	Güncel Haberlerin Aynı Haber Programında Tekrar İşlenmesi	45
5.	SONUÇ.....	46
	KAYNAKLAR.....	48
	EKLER	50
	Ek 1 UML Diagramları.....	50
	ÖZGEÇMİŞ.....	57
	Başarılar.....	57

ŞEKİL LİSTESİ

	Sayfa
Şekil 1.1	Video görüntü bileşenleri1
Şekil 2.1	Sistem tasarımı5
Şekil 2.2	Kenar piksellere göre sahne geçişi yakalama7
Şekil 2.3a	Gerçek çerçeve resmi.....9
Şekil 2.3b	Görüntüye 3x3 Gauss maskesi uygulanmış resim..9
Şekil 2.3c	Görüntüye 5x5 Gauss maskesi uygulanmış resim.....9
Şekil 2.3d	Görüntüye 7x7 Gauss maskesi uygulanmış resim.....9
Şekil 2.4a	x yönündeki kenar değişimlerini10
Şekil 2.4b	y yönündeki kenar değişimlerini yakalayan Sobel maskesi bileşenleri10
Şekil 2.5	Sobel maskesi uygulandıktan sonra oluşan resim10
Şekil 2.6	Renk değişimi açıları.....11
Şekil 2.7	En büyük olanı bastırma işlemi sonucu oluşan resim11
Şekil 2.8	Çift eşik seviye uygulanmış resim12
Şekil 2.9	Başarılı tanınan yüz örnekleri15
Şekil 2.10	Tanınamayan yüz örnekleri15
Şekil 2.11	İçeriği farklı fakat yazı bölgesi aynı iki çerçeve örneği16
Şekil 2.12	Bimodel renk parlaklık dağılımı17
Şekil 3.1	Sahne geçişi olmayan örnek çerçeveler.....20
Şekil 3.2	Sahne geçişi olan örnek çerçeveler20
Şekil 3.3ac	Sahne geçişi olmayan çerçeveler.22
Şekil 3.3bd	Çerçevelere ait gri seviyesi histogramları22
Şekil 3.4ac	Sahne geçişi olan çerçeveler.22
Şekil 3.4bd	Çerçevelere ait gri seviyesi histogramları22
Şekil 3.5	Sahne geçişi olmayan ardışık spiker çerçeveleri.....24
Şekil 3.6	Çerçevelere kenar belirleme yöntemi uygulanmış resimler24
Şekil 3.7	Çerçevelere genişletme ve ters çevirme işlemi uygulanmış resimler.....24
Şekil 3.8	Çerçevelere VE işlemi uygulanması sonucu oluşan resimler25
Şekil 3.9	Sahne geçişi olan örnek ardışık çerçeveler.....25
Şekil 3.10	Çerçevelere kenar belirleme yöntemi uygulanmış resimler26
Şekil 3.11	Çerçevelere genişletme ve ters çevirme işlemi uygulanmış resimler.....26
Şekil 3.12	Çerçevelere VE işlemi uygulanması sonucu oluşan resimler26
Şekil 3.13	Sağdaki spiker için FAR ve FRR grafiği30
Şekil 3.14	Sağdaki spiker için FAR ve FRR grafiği30
Şekil 3.15	Sağdaki spiker için FAR ve FRR grafiği31
Şekil 3.16	Sağdaki spiker için FAR ve FRR grafiği31
Şekil 3.17	Sağdaki spiker için FAR ve FRR grafiği32
Şekil 3.18	Yazı analizi için örnek çerçeveler33
Şekil 3.19	Yazı bölgeleri33
Şekil 3.20	OTSU sonucu elde edilen siyah-beyaz resimler33
Şekil 4.1	Kanal-1 haberi örnek çerçeveleri37
Şekil 4.2	Kanal-2 haber örnek çerçeveleri.....38
Şekil 4.3	Kanal-2 haber örnek 2 çerçeveleri.....40
Şekil 4.4	Kanal-3 haber örnek çerçeveleri.....41
Şekil 4.5	Kanal-3 haber örnek 2 çerçeveleri.....42
Şekil 4.6	Yüz tanımanın başarısız olduğu çerçeve örnekleri.....44
Şekil 4.7	İki haber arasında haber videosu olmayan örnek çerçeveler.....45
Şekil 4.8	Haber özetlerine örnek ardışık çerçeveler45

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 3.1	Sahne geçişi olmayan örneklerde ardışık çerçevelerde mesafe.....21
Çizelge 3.2	Sahne geçişi olan örneklerde ardışık çerçevelerde mesafe21
Çizelge 3.3	Sahne geçişi olmayan örneklerde kenar piksel değişimi mesafe dağılımı23
Çizelge 3.4	Sahne geçişi olan örneklerde kenar piksel değişimi mesafe dağılımı23
Çizelge 3.5	K değerlerine göre mesafe hesabı 128
Çizelge 3.6	K değerlerine göre mesafe hesabı 228
Çizelge 3.7	K değerlerine göre mesafe hesabı 328
Çizelge 4.1	Örnek video özellikleri ve açıklamaları35
Çizelge 4.2	Analiz sonucu çıkan değerler ve açıklamaları.....36
Çizelge 4.3	Kanal-1 örnek video özellikleri.....37
Çizelge 4.4	Kanal-1 örneği analiz sonuçları.....37
Çizelge 4.5	Kanal-2 örnek 1 video özellikleri38
Çizelge 4.6	Kanal-2 örneği 1 analiz sonuçları.....39
Çizelge 4.7	Kanal-2 örnek 2 video özellikleri40
Çizelge 4.8	Kanal-2 örneği 2 analiz sonuçları.....40
Çizelge 4.9	Kanal-3 örnek 1 video özellikleri.....41
Çizelge 4.10	Kanal-3 örneği 1 analiz sonuçları.....42
Çizelge 4.11	Kanal-3 örnek 2 video özellikleri43
Çizelge 4.12	Kanal-3 örneği 2 analiz sonuçları.....43

KISALTMA LİSTESİ

PCA	Principal Component Analysis
SVM	Support Vector Machine
FPS	Frames per second
T	Threshold
FA	False Acceptance
FR	False Rejection
TA	True Acceptance
TR	True Rejection
FAR	False Acceptance Rate
FRR	False Rejection Rate

ÖNSÖZ

Haber programlarının bölütlenmesini hedef alan ve başarılı ayrıştırmanın nasıl yapılmasını sunan bu çalışma Yıldız Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü Mühendislik Yüksek Lisans Programı'nda yürütülmüştür.

Tez çalışmamın ortaya çıkmasındaki ve hazırlanmasındaki katkılarından dolayı tez danışmanım sayın Yard. Doç. Dr. M. Elif Karşılıgil'e ve beni yalnız bırakmamakla kalmayıp, çalışmamda her türlü desteği veren aileme ve en yakın arkadaşım Azer Abizade'ye teşekkür ederim.

İstanbul, Haziran 2009

Rajab Davudov

ÖZET

Bu çalışmada video görüntülerini değerlendirerek haber programlarının haber geçişlerine göre bölütlenmesini sağlayan bir sistem tasarlanmış ve gerçekleştirilmiştir. Türkiye televizyonlarındaki haber programlarının genel karakteristiği spikerin haber hakkında bilgi vermesinin ardından habere ait görüntülerin gösterilmesi, bunun ardından aynı haberle ilgili başka bir alt konu için veya yeni bir haber ile ilgili açıklama yapmak üzere tekrar spikerin görüntülenmesidir. Bundan başka, spikerin görüldüğü çerçevelerde her kanal için ekranın alt bölümünde, yeri sabit olacak şekilde haber başlığının yer aldığı görülmüştür. Geliştirilen sistem, haberlerin bu yapısına göre haber geçişlerini bulabilmeyi hedeflemektedir.

Sistem katmanlar halinde tasarlanmıştır. İlk katmanda ardışık video çerçevelerinin benzerliğini değerlendirmek için renk ve kenar piksel değişimlerinin oranına bakılmıştır. Benzer olan çerçeveler aynı sahneye ait kabul edilerek gruplanmıştır. İkinci katmanda, ilk katmanda elde edilen her sahne için yüz tanıma işlemi uygulanarak spiker olan sahneleri bulunmuştur. Bir sahne içindeki çerçeveler birbirine benzediği için, çerçevelerin tamamına bakmak yerine, sadece ilk N tanesinde spiker yüzü tanıma işlemi yapılarak yüz tanıma işlem zamanından kazanılmıştır. Son katmanda, habere ait video görüntülerinden tekrar spiker görüntüsüne geçildiğinde aynı haberin devam edip etmediğinin anlaşılması için daha önceden işaretlenmiş olan yazı bölgesindeki yazı benzerliklerine bakılarak ardışık çerçevelerin aynı habere ait olup olmadığına karar verilmiştir.

Farklı aşamalarda, ardışık sahneler arasında ortak özellikler bulunduğunda bu sahneler birleştirilerek tek sahne haline getirilmiştir. Bu şekilde haber programının haber geçişlerine göre bölümlere ayrılması işlemi gerçekleştirilmiştir. Geliştirilen sistemin haber geçişlerini bulma başarısı farklı haber kanallarından alınan örnek haber programları üzerinde denenerak değerlendirilmiştir. Sistemin haber geçişlerini doğru bulma başarısının ortalama %99 olduğu görülmüştür.

Anahtar Kelimeler: Video özetleme, Sahne geçişi yakalama, Haber videoları

ABSTRACT

This article is about segmenting broadcast news program into individual news. In Turkish television news programs, anchors first give brief information about the news itself and then news related video is shown. Next time anchor is seen, he talks about different news or some other sub news related to the previous one. During anchor scenes there is news related title on the fixed location of the screen. This information is used by the system to find scene cuts.

Segmentation system designed in this project consist of several layers. First one analyzes frames based on the change ratios in their color histogram and edge pixel. Similar frames are marked as a same scene. Next layer aims to find anchors in scene that are marked by the first layer. If anchor is found in N frames, then in order to increase the speed of the analysis, the system stops there. Third layer analyzes news title regions in the frames in order to understand if two different scenes are related to the same news or not. If two scenes have similar characteristics they are merged into the same scene. Last layer merges anchor scenes with non-anchor scenes to form final segmented news scene of the news program.

In this project, scenes with common properties were merged for different layers. By this our segmentation process was completed. Designed system was tested for different news programs and 99% success rate in segmentation was recorded.

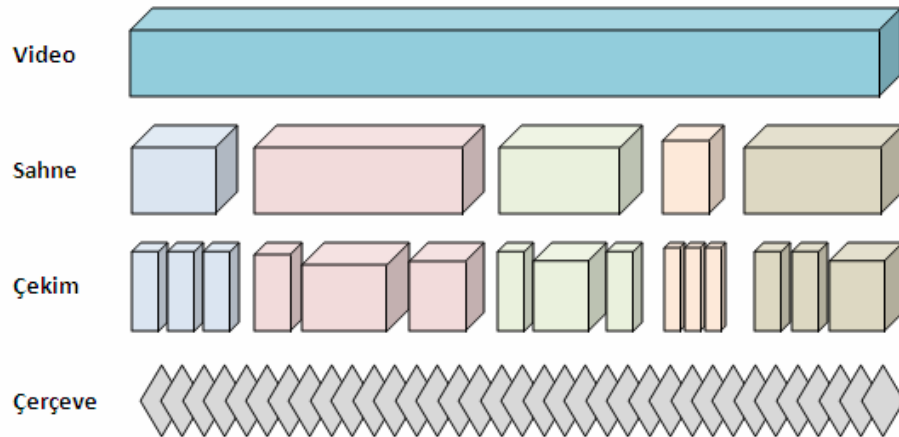
Keywords: Video segmentation, Scene detection, News videos

1. GİRİŞ

İnternet olanaklarının gelişmesi ve yaygınlaşması bilgisayar dünyasındaki bir çok alanın da gelişmesine katkıda bulunmuştur. Kullanıcıların yaygın veri paylaşımı sonucunda sanal alemde çok önemli bir veri deposu oluşmaktadır. Bu verilerin önemli bir kısmı çoğulortam olarak isimlendirilen hem görüntü, hem ses, hem de yazı verilerinin bir araya gelerek oluşturduğu hibrid bilgilerdir.

Video görüntülerin yaygınlaşması ve paylaşılması video analizi konusundaki çalışmaların önem kazanmasına sebep olmuştur. Bu verileri işleyip anlamlı sonuçlar çıkarmaya yönelik sistemler tasarlanmaya başlanmış ve bu konuda bir çok araştırma ve uygulama geliştirilmiştir. Daha önceleri sabit sayısal görüntüler üzerinde yapılan bir çok araştırma, video görüntüler için daha etkin hale getirilerek kullanılmaya başlanmıştır. Bir video bir saniyede ardışık 25-30 sabit resimden oluşan görüntü topluluğudur. Dolayısıyla video analizi, aslında bir resim yerine binlerce resim üzerinde işlem yaparak ve bu resimlerin zamana göre birbirleri ile olan bağlantılarını incelemekten ibarettir.

Video görüntüsü çerçevelerden, çekimlerden ve sahnelerden oluşmaktadır. Her video çerçevesi bir resimdir. Çekim bir kameranın kayıt işlemine başlamasından, başka bir kameranın kayıt işlemine başlamasına kadar geçen sürede elde edilen çerçevelerdir. Nesne, insan, mekan ve zaman bakımından anlamsal benzerliğe sahip çekimlerin birleşmesi ile sahneler oluşmaktadır. Bütün sahneler ise video görüntüsünün tamamını oluşturmaktadırlar. Vide görüntüsünün bileşenleri Şekil 1.1’de görülebilir. [1]



Şekil 1.1: Video görüntü bileşenleri [1]

Televizyon kanallarında hergün farklı saatlerde haber programları yayınlanmaktadır. Haberler farklı konularda bir çok olayla ilgili önemli bilgiler içerir. Bu veriler şu anda önemli olduğu gibi ilerki zamanlarda da gerek referans olmak, gerekse de bilgi amaçlı erişilmesi istenilebilecek bilgilerdir. Bu kadar karmaşık ve çok büyük hacimli video arasından istenilen bilgiye ulaşmak bir hayli zor ve zaman alıcıdır. Bu duruma çözüm bulmak için, bu verinin en efektif şekilde nasıl saklanabileceği belirlenmelidir. Bunun için öncelikle bir bütün olan haber programları haberlere göre bölünerek etiketlenmelidir. Daha sonra elde edilen haber parçaları içeriklerine göre etiketlenebilir, sıkıştırma yöntemleri kullanılarak arşivlenebilir ve istendiğinde verilen etiketlere göre erişilip izlenebilir.

Bu çalışmada haber programı videolarının anlamlarına göre bölütlenmesini sağlayan bir sistem tasarlanmış ve gerçekleştirilmiştir. Önce haber videosu çerçevelere ayrılır. Bunu yapmak için her video çerçevesi farklı bir dosya olarak kaydedilir. Daha sonra ardışık çerçeveler arasındaki benzerlikler incelenerek çekimler bulunur. Bulunan çekimler ortak özelliklerine göre birleştirilerek sahneler haline getirilir. Sahne geçişlerinde yüz tanıma ile spiker olup olmadığına bakılır. Spiker bulunan çerçeveler haber geçişi olarak işaretlenir. Daha sonra çerçevedeki yazı bölgesi benzerlikleri değerlendirilerek haber geçişi olup olmadığı kesinleştirilir. Analiz sonucunda haber videosunda anlatılan haberlerin başlangıç ve bitiş süreleri bulunmuş olur.

1.1. Önceki Çalışmalar

Haber programlarının özetlenmesi ve haber programlarından bilgi çıkartılmasına yönelik pek çok çalışma mevcuttur. Haber programı bölütlemesi için S.R.Maskey ve J. Hirschberg tarafından yapılan bir çalışmada Bayes Ağları kullanılmıştır. İlk önce her programda çıkarılacak özellikler belirlenmiş daha sonra bu özelliklere göre bir Bayes Ağı tasarlanmıştır. Çıkarılan özellikler sahnenin tüm video içerisindeki yeri, uzunluğu ve içeriğinde spikerin olmasıdır. Bundan başka sahneden önceki ve sonraki sahnelerin de uzunluğu ve spiker özellikleri çıkarılmıştır. Kısa süren sahneler ve başta veya sonda olan sahneler daha az önemli, uzun süren ve ortalarda olan sahneler ise daha önemli olarak işaretlenmiş ve Bayes Ağı üzerinden özetleme yapılmıştır [2].

N. O'Connor, C. Czirjek, S. Deasy, S. Marlow, N. Murphy ve A. Smeaton tarafından yapılan çalışmada haber programlarının haberlere bölütlenmesi hedef alınmıştır. Bu çalışmadaki yaklaşımda ilk önce renk farklılıklarına bakılarak sahne geçişleri yakalanmıştır. Daha sonra

benzer renk özelliklerine sahip sahneler birleştirilir. Sonraki aşama olarak sahnelerde spiker aranır. Spiker aramak için çerçevede yüzler bulunur. Çerçeve bir arka plan ve en fazla üç yüze sahip ise spiker çerçevesi olarak algılanır. Bu bağlantılar çıkarıldıktan sonra farklı spiker sahneleri birbirleriyle karşılaştırılır. Eğer sahnelerin benzerlikleri 98% üzerinde ise aynı habere ait oldukları kabul edilir aksi takdirde farklı haberler olarak işaretlenir. Çalışmanın bölütleme başarısı 84% olarak gösterilmiştir [3].

Diğer bir çalışma haber videolarını Destek Vektör Makinaları (Support Vector Machine-SVM) kullanılarak sınıflandırmayı hedef almıştır. Bu çalışmada ilk olarak görüntü, renk özelliklerine göre sahnelere bölünür. Daha sonra sahnelerdeki yazılar çıkarılır. Yazıdaki kelimeler filtreden geçirilerek çok kullanılan kelimeler metin bilgisinden çıkartılır ve geriye kalan bilgi üzerinden SVM kullanılarak sınıflandırma yapılır [4].

İçeriğe bakılarak bölütleme çalışmaları daha başarılı sonuçlar vermektedir. İçeriği değerlendiren bir çalışmada haber videosundaki sahneler üç sınıfa ayrılmıştır:

- Spiker konuşması
- Muhabir videosu
- Röportaj

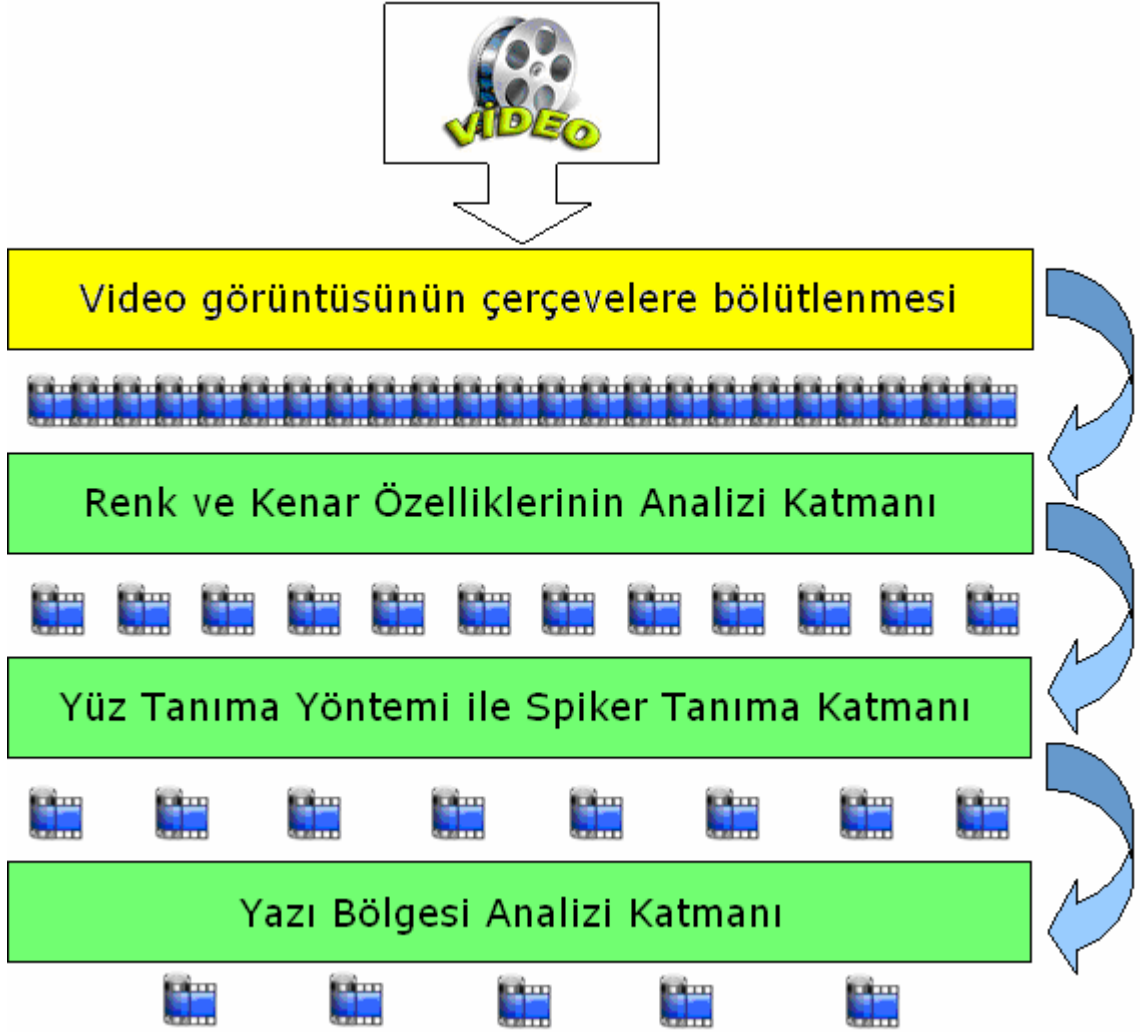
Bu sınıflar için bir Sonlu Durum Makinesi (Finite State Machine) tasarlanmıştır. Daha sonra her sahne özelliklerine göre bu makinada farklı durumlara geçilmesini tetiklemiştir. Bunun yanı sıra sahneler içeriğine göre politika, finans, spor vs. olarak sınıflandırılmıştır [5].

Bazı çalışmalarda ise görüntü özelliklerinin yanı sıra sinyal işleme ile anlatılan haberleri ses olarak özellikleri çıkarılmış ve daha sonra bu özellikler ile sahnelerin benzerlikleri çıkarılarak haber videosu haberlere bölütlenmiştir. M. Bertini, A. Del. Bimbo ve P. Pala tarafından yapılan çalışmada IBM ViaVoice'98 ses tanıma sistemi kullanılmıştır. Bu çalışmada Saklı Markov Modeli(Hidden Markov Model) yöntemi ile ses bilgisinin içeriği analiz edilmiştir. Analiz sonucunda haber içeriği spor, ekonomi, politika vs. şekilde sınıflandırılmıştır. Bu çalışmada spiker sesi önceden eğitilerek sadece spiker konuşması üzerinde analiz yapılması sağlanmıştır. Fakat görüntü özellikleri bölütleme için oldukça yeterli olduğu için ses özellikleri daha çok sahne içeriğini anlamak ve sınıflandırmak için kullanılmıştır. [6]

2. HABER PROGRAMI BÖLÜTLEME SİSTEMİ TASARIMI

Bu çalışmada, bir haber programını haber geçişlerine göre bölümlere ayıran bir sistem tasarlanmış ve gerçekleştirilmiştir. Haber geçişlerini bulabilmek için haber videolarının karakteristik özellikleri farklı yöntemlerle değerlendirilmiştir. Hız ve başarıyı arttırmak için bütün özellikleri birlikte değerlendirmek yerine dört katmanlı bir sistem tasarlanmıştır (Şekil 2.1). Haber programlarının genel karakteristiği spikerin haber hakkında bilgi vermesinin ardından habere ait görüntülerin gösterilmesi, bunun ardından aynı haberle ilgili başka bir alt konu için veya yeni bir haber ile ilgili açıklama yapmak üzere tekrar spikerin görüntülenmesidir. Buna göre spikerin konuştuğu bölümler haber değişimlerini ayırmak için kullanılabilir. Bunun için ilk katmanda ardışık görüntülerin renk ve kenar değişimi özellikleri değerlendirilerek az değişimli görüntüler elenmiştir. Kalan görüntüler üzerinde spiker aranarak haber geçişleri bulunmuştur. Sistemin son katmanı olan yazı analizi bölümünde haberle ilgili görüntüde yer alan yazılar incelenerek benzerlikler bulunmuştur. İlk iki katmanın yetersiz kaldığı durumlarda son katman sayesinde başarılı bir bölütleme ortaya konmuştur.

Her katmana giriş olarak bir önceki katman sonucunda bulunan sahne bilgileri aktarılmaktadır. Sahne nesne, insan, mekan ve zaman bakımından anlamsal benzerliğe sahip çekimlerin birleşmesidir. Sahne bilgisi başlangıç ve bitiş çerçeve numarasından oluşmaktadır. İlk katmanın giriş bilgisi bütün çerçevelerdir. İlk katman, çerçeveleri sahne geçişi bulma yöntemleri kullanarak, sahneler haline getirir. Diğer katmanlar analiz işlemlerini sahneler üzerinde yapar. Sahnedeki çerçeveler işlenirken istenilen sonuçlar elde edildiğinde kalan çerçevelere bakılmadan bir sonraki sahneye geçilir. Örneğin, bir sahnede spiker aranıyorsa, ilk N tane çerçevede spiker tanındıktan sonra diğer çerçevelere bakılmadan bir sonraki sahneye geçilir. Katmanlar analiz işlemlerini yaparken saheneler arası ortak özellikler bulunduğunda bu sahneleri birleştirilerek tek sahne haline getirir. Dolayısıyla girişte N tane sahne varsa, çıkışta M tane sahne oluşur ve her zaman $M \leq N$ 'dir. Ardışık olan iki sahnede, ilk sahnede spiker görülüyorsa ve sonraki sahne başka bir video görüntü ise bu iki sahne gerçekte aynı habere ait görüntülerdir. Bu iki sahne birleştirilerek tek bir sahne haline getirilir.



Şekil 2.1: Sistem tasarımı

2.1. Renk ve Kenar Özelliklerinin Analizi Katmanı

Bu katmanda temel görüntü işleme yöntemleri kullanılarak, içerik analizi yapılmadan sahneler arası geçişler bulunmaktadır. Sadece bu katmanın giriş verisi sahne listesi yerine bütün çerçevelerdir. Bu katman her çerçeve için bir sonraki çerçeveye bakarak çekimleri bulmaya çalışır. Çekimler bir kameranın kayıt işlemine başlamasından, başka bir kameranın kayıt işlemine başlamasına kadar geçen sürede elde edilen çerçevelerdir. Bunun için önce histogram farklarına bakılır [7]. Eğer fark, belirli eşik seviyesinden yüksek ise geçiş olarak işaretlenir, değilse kenar pikselleri farkına bakılır. Bu analizin sonucu da belirli eşik seviyesinden yüksek ise geçiş belirlenmiş olur. Bu katman sonucunda haber videosundaki bütün çekimler bulunarak bir sonraki katmana iletilir.

2.1.1. Renk Değişiminin Analizi

Ardışık iki çerçevede değişiklik olup olmadığını anlamanın en temel yolu renk değişimlerini değerlendirmektir. Bu nedenle haber geçişlerini bulmak için ilk olarak renkli histogram farkları değerlendirilmiştir. Bunun için RGB uzayında her bileşenin kullanım olasılıkları ayrı ayrı değerlendirilmiştir. NXM büyüklüğündeki renkli bir resmin histogramının bulunmasını gösteren yarı-kodlama aşağıdaki gibidir.

```

1. HISTOGRAM_RED = ARRAY OF INTEGER [0..255]
2. HISTOGRAM_GREEN = ARRAY OF INTEGER [0..255]
3. HISTOGRAM_BLUE = ARRAY OF INTEGER [0..255]
4. FOR x = 1 .. N DO
5.     FOR y = 1 .. M DO
6.         r = IMAGE[x, y].RED ;
7.         g := IMAGE[x, y].GREEN ;
8.         b := IMAGE[x, y].BLUE ;
9.         HISTOGRAM_RED[r] := HISTOGRAM_RED[r] + 1 ;
10.        HISTOGRAM_GREEN[g] := HISTOGRAM_GREEN[g] + 1 ;
11.        HISTOGRAM_BLUE[b] := HISTOGRAM_BLUE[b] + 1 ;

```

Ardışık iki çerçevenin renk değişimlerine göre benzerliğinin ölçümünde, her bileşen için iki çerçeveye ait histogram değerlerinin arasındaki mesafe Öklit Mesafesi kullanılarak hesaplanmıştır. (Eşitlik 2.1). Burada R_1 , G_1 , ve B_1 ilk çerçeveye ait RGB bileşenlerinin histogram değerlerini, R_2 , B_2 ve G_2 ise ardından gelen çerçeveye ait RGB bileşenlerinin histogram değerlerini göstermektedir.

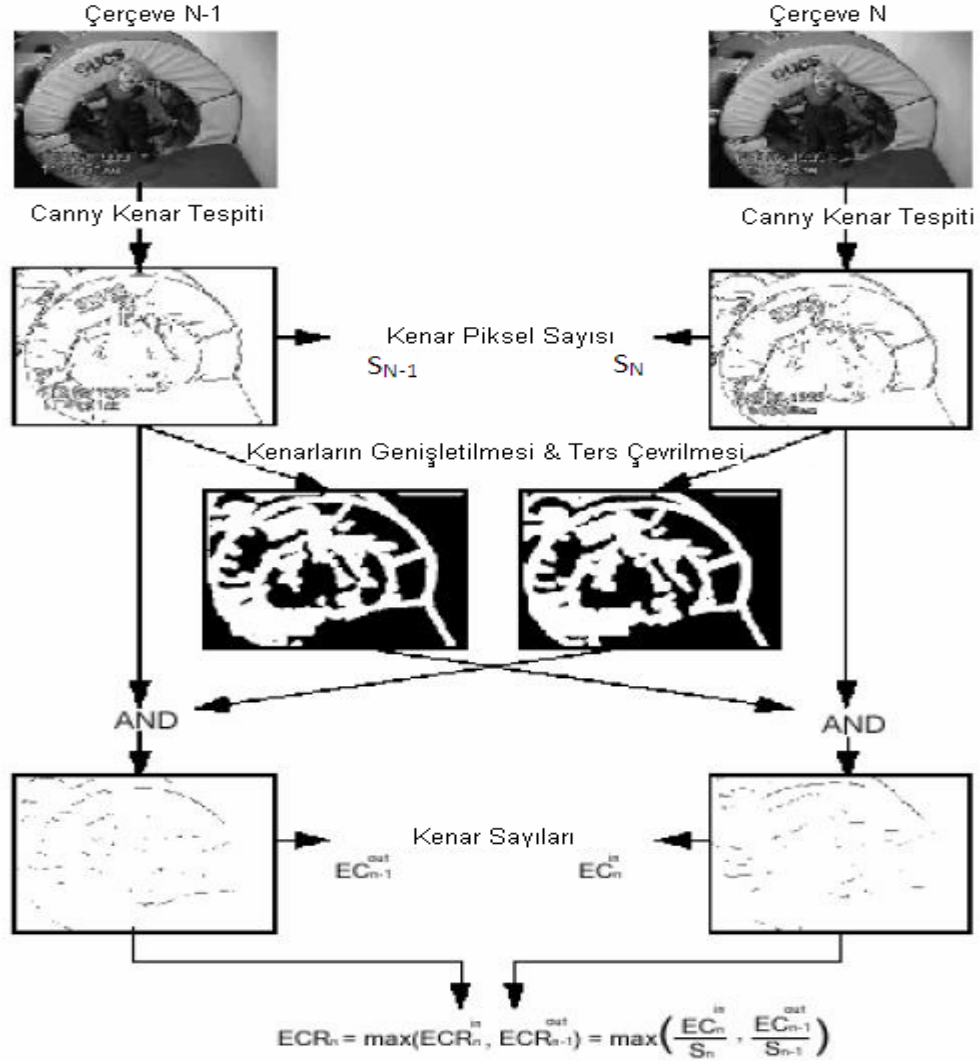
$$Mesafe = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (2.1)$$

Bulunan mesafe belirlenmiş eşik seviyesinden büyük ise iki çerçeve arasında belirgin bir renk değişimi olduğu için içerik değişimi olduğu söylenebilir. Eşik seviyesi deneysel sonuçlar ile belirlenmiştir.

2.1.2. Kenar Değişimi Analizi

Haber geçişlerini bulmak için renk değişimlerini değerlendirmek tek başına yeterli olmamaktadır. Ardışık görüntülerde renk değişimi az olmasına rağmen görüntü içeriğinde de önemli değişiklikler olabilmektedir. Bunun için renk benzerliklerine göre farklılık

göstermeyen ardışık görüntülerin kenar piksellerinin değişimi değerlendirilmiştir. Şekil 2.2’de bu yöntemin işlem adımlarının blok diyagramı görülmektedir.



Şekil 2.2: Kenar piksellere göre sahne geçişi yakalama [10]

Bu yöntemde ilk önce çerçevelere kenar piksel bulma işlemi uygulanır. Daha sonra oluşan ikili resimlere genişletme ve ters çevirme işlemlerini uygulayarak iki çerçeve arasındaki farklara bakılır[10]. Ardışık N-1 ve N. Çerçevler için yöntemin aşamaları aşağıdaki gibidir:

- ```
// Çerçevelere Canny kenar belirleme yöntemi uygulanır.
1. IMAGE_CANNY[N-1] := CANNY(IMAGE[N-1]) ;
2. IMAGE_CANNY[N] := CANNY(IMAGE[N]) ;
// Canny sonucunda oluşan ikili resimlere genişletme ve ters çevirme işlemleri uygulanır.
3. IMAGE_BINARY[N-1] := INVERSION(DILATION(IMAGE_CANNY[N-1])) ;
4. IMAGE_BINARY[N] := INVERSION(DILATION(IMAGE_CANNY[N])) ;
// Oluşan resimler Canny sonucundaki resimlerle AND ile çapraz olarak çarpılır.
5. IMAGE_AFTERAND[N-1] := IMAGE_CANNY[N-1] AND IMAGE_BINARY[N] ;
```

```

6. IMAGE_AFTERRAND[N] := IMAGE_CANNY[N] AND IMAGE_BINARY[N-1] ;
// Canny sonucundaki kenar piksel sayıları hesaplanır.
7. S[N-1] := IMAGE_CANNY[N-1].PIXEL_COUNT ;
8. S[N] := IMAGE_CANNY[N].PIXEL_COUNT ;
AND işleminin sonucundaki piksel sayıları hesaplanır.
9. EC[N-1] := IMAGE_AFTERRAND[N-1].PIXEL_COUNT ;
10. EC[N] := IMAGE_AFTERRAND[N].PIXEL_COUNT ;
// Bu iki piksel sayıları oranlanır.
11. ECRATIO[N-1] := EC[N-1] / S[N-1] ;
12. ECRATIO[N] := EC[N] / S[N] ;
//En büyük oran mesafe olarak alınır.
13. ECDISTANCE := MAX(ECRATIO[N-1], ECRATIO[N]) ;

```

Aşağıdaki alt başlıklarda bu işlem adımlarında kullanılan yöntemler anlatılmaktadır.

### 2.1.3. Canny Kenar Belirleme Yöntemi

Kenarlar, görüntüdeki nesneleri bir birinden ayıran piksellerdir. Kenar piksellerin çevrelerindeki piksellerden ayırt edilmesini sağlayan, renklerdeki belirgin farklılıktır. Kenar piksellerin yerini belirlemek için kullanılan yöntemlerden en başarılısı içerdiği alt adımların özelliklerinden ötürü Canny Kenar Belirleme yöntemidir [11].

Canny Kenar Belirleme yönteminin diğer yöntemlere göre farkı zayıf ve güçlü kenar piksellerini ayırarak zayıf kenar piksellerini elemesi ve çift eşikdeğeri uygulaması sayesinde daha başarılı bir şekilde kenar pikselleri bulmasıdır. Canny Kenar Belirleme aşağıdaki adımlardan oluşmaktadır[11]:

1. Resim Gauss filtresi ile yumuşatılır.
2. Her pikselin kenar olma oranı hesaplanır.
3. Aynı yöndeki komşularından büyük olmayan kenar pikseller elenir (Non-maximum suppression).
4. Çift eşik seviyesi uygulanır (Hysteresis thresholding).

#### 2.1.3.1. Resmin Gauss Filtresi ile Yumuşatılması

Gürültü piksellerinin çevrelerindeki piksellere göre renk olarak keskin farkı olduğu için kolaylıkla kenar pikseli olarak algılanabilir. Bu durumu ortadan kaldırmak için bu pikseller Gauss filtresi ile yumuşatılmalıdır(Eşitlik 2.2). Filtre olarak Eşitlik 2.3-deki maske kullanılmıştır.

$$g(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.2)$$

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A \quad (2.3)$$

Eşitlik 2.3-deki A kaynak görüntü, B ise Gauss uygulanmış görüntüyü göstermektedir.

5x5 maske kullanılmasının sebebi 3x3 maskenin gürültüyü yok etmekte yetersiz olması, 7x7 maskenin ise resmin bulanıklaşmasına sebep olmasıdır. Bu durumu Şekil 2.3-te görebiliriz.



(a)



(b)



(c)



(d)

**Şekil 2.3: a) Gerçek çerçeve resmi. b) Görüntüye 3x3 Gauss maskesi uygulanmış resim. c) Görüntüye 5x5 Gauss maskesi uygulanmış resim. d) Görüntüye 7x7 Gauss maskesi uygulanmış resim.**

### 2.1.3.2. Sobel Maskeleri ile Kenar Piksellerin Değer ve Yönünün Belirlenmesi

İkinci aşamada her piksele Sobel maskeleri(Şekil 2.4) uygulanarak piksel bazında kenar olma eğilimi ve renk değişiminin yönü hesaplanır. Sobel maskeleri hem x hem de y eksenini yönünde uygulananırlar. Bu şekilde her yöndeki kenar değişimleri belirlenmiş olur. [1]

|    |   |    |
|----|---|----|
| -1 | 0 | +1 |
| -2 | 0 | +2 |
| -1 | 0 | +1 |

G<sub>x</sub>

(a)

|    |    |    |
|----|----|----|
| +1 | +2 | +1 |
| 0  | 0  | 0  |
| -1 | -2 | -1 |

G<sub>y</sub>

(b)

Şekil 2.4: (a) x yönündeki kenar değişimlerini (b) y yönündeki kenar değişimlerini yakalayan Sobel maskesi bileşenleri

Şekil 2.4-deki G<sub>x</sub> maskesi X yönündeki, G<sub>y</sub> maskesi ise Y yönündeki renk değişimlerini hesaplamak için uygulanır.

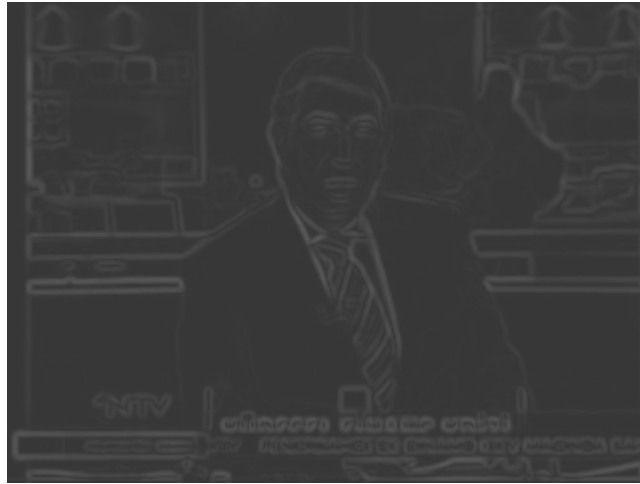
$$\Delta X = S[i, j] * G_x \quad (2.4)$$

$$\Delta Y = S[i, j] * G_y \quad (2.5)$$

Görüntüdeki her piksel(S[i,j]) için ilk önce Eşitlik 2.4'deki konvolüsyon işlemi uygulanarak X yönündeki değişim hesaplanır. Daha sonra Eşitlik 2.5 uygulanarak Y yönündeki değişim hesaplanır. Kenar olma eğilimi  $\Delta X$  ve  $\Delta Y$  karelerinin karekökünü alarak bulunur(Eşitlik 2.6). Kenarın yönünü bulmak için Eşitlik 2.7 kullanılır.

$$\text{Kenar Olma Eğilimi} = \sqrt{\Delta X^2 + \Delta Y^2} \quad (2.6)$$

$$\text{Yön} = \text{Arctan}\left(\frac{\Delta X}{\Delta Y}\right) \quad (2.7)$$

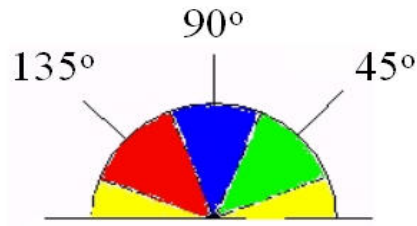


Şekil 2.5: Sobel maskesi uygulandıktan sonra oluşan resim

### 2.1.3.3. En Büyük Olmayanı Bastırma (Non-Maximum Suppression)

Kenar pikselleri mümkün olduğunca hassas belirleyebilmek için, her piksel kenar yönü yönündeki komşuları ile karşılaştırılarak kenar olma eğilimi değerlendirilir. Eğer herhangi bir komşusundan küçük veya eşitse yeteri kadar güçlü bir kenar pikseli olmadığı için değeri sıfır yapılır. Komşularından büyükse değeri korunur. [11]

Bir pikselin kenar yönündeki komşularını belirlemek için kenar yönünü gösteren açı değerleri  $45^\circ$  ve katlarına dönüştürülmelidir. Kenar yön açıları  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  ve  $135^\circ$  olarak belirlenir. Açı değerinin hangi aralığa denk geldiği Şekil 2.6'de görüldüğü şekilde belirlenir.



Şekil 2.6: Renk değişimi açıları

Zayıf olan kenar piksellerin elenmesi ile kenar kalınlığı incelmış ve kuvvetli kenarlar korunduğu için daha belirgin hale gelmiş olur (Şekil 2.7).



Şekil 2.7: En büyük olanı bastırma işlemi sonucu oluşan resim

### 2.1.3.4. Çift Eşik Seviyesi (Hysteresis Thresholding) Uygulanması

Kenar belirleme işlemi sonunda piksel değerleri 0-255 arasında değiştiğinden, hangi piksellerin kenar piksel olması gerektiğine karar vermek için çift aşamalı eşik seviyesi uygulanır. Eğer yüksek bir eşik seviyesi belirlenirse kenar olan piksellerin bazıları elenmiş olabilir. Düşük bir eşik seviyesi belirlenirse de kenar olmayan bir çok gürültü pikselleri

kenar piksel olarak kabul edilebilir. Bu sorunu ortadan kaldırmak için çift eşik seviyeli belirleme yapılır. Bunun için  $T_{yüksek}$  ve  $T_{düşük}$  olmak üzere iki eşik seviyesi belirlenir. Her piksel önce  $T_{yüksek}$  eşik seviyesi ile karşılaştırılır. Eğer  $T_{yüksek}$ 'den büyük ise kenar piksel olarak işaretlenir. Daha sonra komşu piksellere bakılır. Komşu piksel  $T_{yüksek}$ 'den büyük ise veya  $T_{yüksek}$ 'den küçük fakat  $T_{düşük}$ 'den büyük ise kenar piksel olarak işaretlenir. Bütün komşu pikseller işlendikten sonra diğer kenar piksellere geçilir. [11] Şekil 2.8'de çift eşik seviyesi uygulanmış resime örnek gösterilmektedir.



Şekil 2.8: Çift eşik seviye uygulanmış resim

## 2.2. Spiker Tanıma Yöntemi İle Sahne Geçişlerinin Bulunması Katmanı

Renk ve kenar değişimlerini değerlendirerek yapılan eleme ile birbirine benzer, yani aynı sahneye ait ardışık çerçevelerin bulunması sağlanır. Fakat bu özellikler haber geçişlerini yakalamak için yeterli değildir. Haber programlarında genel yapı, spikerin konuşması, haber ile ilgili video görüntülerinin gösterilmesi ve yeni haber için tekrar spiker görüntüsüne geçilmesidir. Bu durumda görüntüde bir insan yüzü görülüyorsa ve bu yüz haber spikerine aitse, haber spikeri başarı ile tanınırsa haber geçişlerini yakalamak mümkün olacaktır.

Bu katmanda aynı habere ait olan çekimler birleştirilerek tek bir sahne haline getirilir. İlk önce çekimler S ve V şeklinde etiketlenir:

- S: Spikerin bulunduğu sahne
- V: Anlatılan haber ile ilgili sahne

Her sahne işlenirken içeriğinde spiker yüzü bulunduğunda S olarak, bulunmadığında ise V olarak etiketlenir. Her S-V görüntü gruplarının tek bir habere ait olması gerektiği düşünülerek

tek bir sahne olarak birleştirilir. Bu işlem sonucunda 1. katmandan gelen görüntü sayısı gruplanarak yeni bir liste oluşturulur ve bu liste bir sonraki katmana iletilir.

Bir çerçevede yüz tanıma yapılabilmesi için yapılması gereken işlemler aşağıdaki alt bölümlerde anlatılmıştır.

### 2.2.1. Yüz Bulma

Yüz özelliklerinden spikeri tanımak için ilk önce görüntüdeki yüzlerin bulunması gerekir. Bunun için OpenCV kütüphanesi kullanılmıştır. OpenCV resimdeki yüzleri bulmak için Haar-benzer özellik çıkarma yöntemini kullanır.[13]

### 2.2.2. Ana Bileşen Analizi Yöntemi İle Yüz Tanıma

Yüz tanıma için Ana Bileşen Analizi( Principal Component Analysis – PCA) yöntemi [8] kullanılmıştır. Sistem önce, PCA ile özellikleri indirgenmiş spiker yüzleri bilgileri kullanılarak eğitilmiştir. Daha sonra farklı haber videoları için, çerçevede bulunan yüzlere PCA uygulanmıştır. Yüzün spiker veritabanındaki en benzer yüze mesafesi belirli bir eşik seviyesinden küçükse yüz spiker yüzü olarak işaretlenmiştir.

Bu yöntemin tanıma ve eğitim aşamaları için işlem adımları şöyledir [9]:

#### Eğitim:

1. Yüz resmindeki pikseller  $M$  uzunluklu bir vektör haline getirilir. Eğitim verisindeki örnekler kullanılarak her özellik için ortalama değerler bulunur (Eşitlik 2.8).

$$\bar{X} = \frac{1}{M} \sum_{i=1}^M X_i \quad (2.8)$$

2. Eğitim setindeki her örneğin  $M$  uzunluklu özellik vektöründen ortalama çıkartılır(Eşitlik 2.9).

$$\phi_i = X_i - \bar{X} \quad (2.9)$$

3. İkinci adımda elde edilen vektörler ( $\phi_i$ ) her vektör bir sütun olacak şekilde bir matrise yerleştirilir. Örneğin  $N$  özellik sayısı ve  $M$  örnek sayısı ise  $N \times M$  şeklinde bir matris (Eşitlik 2.10) oluşur. Bu matrisi kendi transpozitesiyle çarpıldığında özelliklerin birbirine göre değişimini gösteren kovaryans matrisi (Eşitlik 2.11) elde edilir.

$$A = [\phi_1 \phi_2 \dots \phi_M] \quad (2.10)$$

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \quad (2.11)$$

4. Kovaryans matrisinin özdeğerleri (eigenvalue) hesaplanır(Eşitlik 2.12). Bu eşitlikte  $\lambda$  skaları özdeğeri,  $v$  vektörü ise özvektörü (eigenvector) göstermektedir.

$$(A^T A)v_i = \lambda_i v_i \quad (2.12)$$

5. Özdeğerler büyükten küçüğe doğru sıralanır (Eşitlik 2.13).

$$\lambda_1 > \lambda_2 > \dots > \lambda_N \quad (2.13)$$

6. İlk K adet özdeğere karşılık gelen özvektörler alınarak bir matrise yerleştirilir (Eşitlik 2.14) tane eigen vektör değerleri tanımda kullanılır. K değeri deneysel sonuçlar ile belirlenir.

$$\lambda = \begin{bmatrix} \lambda_{11} & \lambda_{21} & \dots & \lambda_{K1} \\ \lambda_{12} & \lambda_{22} & \dots & \lambda_{K2} \\ \vdots & \vdots & \dots & \vdots \\ \lambda_{1M} & \lambda_{2M} & \dots & \lambda_{KM} \end{bmatrix} \quad (2.14)$$

7. İkinci adımda elde edilen vektörler ( $\phi_i$ ) özvektörler ile çarpılarak her örneğin özuzaya (eigenspace) geçmesi sağlanır (Eşitlik 2.15). Elde edilen bu vektörlere özyüzler (eigenfaces) denir. Bu vektörler, örneklere ait indirgenmiş özelliklerdir.

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_K \end{bmatrix} = \lambda^T \times \phi_i \quad (2.15)$$

### Tanım:

1. Yüz resmindeki pikseller  $M$  uzunluklu bir vektör haline getirilir.  $M$  uzunluklu özellik vektöründen ortalama vektörü ( $\bar{X}$ ) çıkarılır (Eşitlik 2.16).

$$\phi_x = X - \bar{X} \quad (2.16)$$

2. İkinci adımda elde edilen vektör ( $\phi_x$ ) özvektörler ile çarpılarak her örneğin özuzaya (eigenspace) geçmesi sağlanır (Eşitlik 2.17).

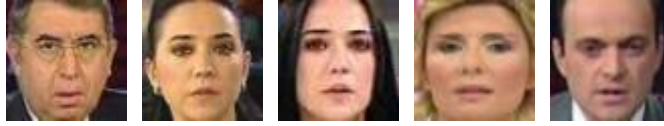
$$\begin{bmatrix} \lambda_{x1} \\ \lambda_{x2} \\ \vdots \\ \lambda_{xK} \end{bmatrix} = \lambda^T \times \phi_x \quad (2.17)$$

3. Eşitlik 2.17’de özuzaya geçmiş olan bilinmeyen örneğin, eğitim veritabanındaki örneklerle mesafesi Öklit Mesafe yöntemi ile ölçülür ve en çok benzediği örnek bulunur (Eşitlik 2.18).

$$D = \sqrt{(\lambda_1 - \lambda_{x1})^2 + (\lambda_2 - \lambda_{x2})^2 + \dots + (\lambda_K - \lambda_{xK})^2} \quad (2.18)$$

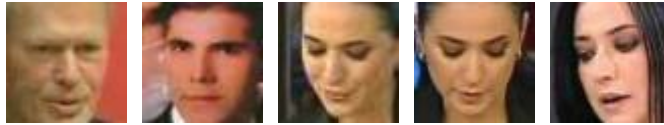
### 2.2.3. Yüz Tanıma Başarısının Değerlendirilmesi

Yüz tanıma başarısı görüntü özelliklerine ve görüntüdeki yüzün yönüne göre değişmektedir. Eğitim verisindeki yüz resimleri devamlı ön taraftan çekilen görüntüler olduğu zaman yan taraftan görülen yüzün tanınması mümkün değildir. Şekil 2.9’de başarılı tanınan spiker yüzleri gözükmemektedir.



Şekil 2.9: Başarılı tanınan yüz örnekleri

Başarısız tanımalarda spiker yüzlerinin kameraya bakış açısının farklı olması, görüntü kalitesinin yetersizliği gibi sebeplerden ötürü spiker tanınamamıştır. Bu resimlere de Şekil 2.10-daki örnekler gösterilebilir.



Şekil 2.10: Tanınamayan yüz örnekleri

Yüz tanıma katmanını analizi zamanı sahne içinde doğru bulunan yüzler varsa, yanlış bulunan yüzlerin olması analizin başarısını etkilememektedir.

### 2.3. Yazı İçeriğinin Analizi Katmanı

Bu katman daha önceki katmanlarda bulunamayan sahneler arası bağlantıları bularak bölütleme işlemini yapmaktadır. Bu katmanda görüntüdeki yazı alanları değerlendirilerek sahneler arasında yazı içeriğine göre benzerlikler yakalanmaktadır. Haber programlarında görüntünün belirli bir bölgesinde haberler ilgili yazılar yer alabilir. Aynı habere ait sahnelerin görüntü özellikleri birbirinden farklı olsa dahi görüntülerdeki yazı bölgelerindeki

içerik aynı ise aynı haberin parçası oldukları anlaşılabilir. Yazı içeriği olarak birbirine benzeyen görüntüler birleştirilerek tek bir sahne haline getirilir.

Sonuç olarak elde edilen sahne grupları, haber videosunda içeriklerine göre ayrılmış haberleri göstermektedir.

### 2.3.1. Yazı Bölgesi Karşılaştırma İşlemi

Haber programlarında, haber ile ilgili bilgi veren yazı bölgesi hep aynı yerdir. Görüntü özelliklerine göre farklı olan sahnelerin aynı habere ait olduğu durumlarda çerçevenin belirli bölgesinde yeralan yazıların ortak olduğu gözlemlenmiştir (Şekil 2.11). Bu yazıların devamlı aynı bölgede yer alması ve değişmeyen renk özelliklerine sahip olması sahneler arası benzerliklerin anlaşılmasını kolaylaştırmaktadır.



Şekil 2.11: İçeriği farklı fakat yazı bölgesi aynı iki çerçeve örneği

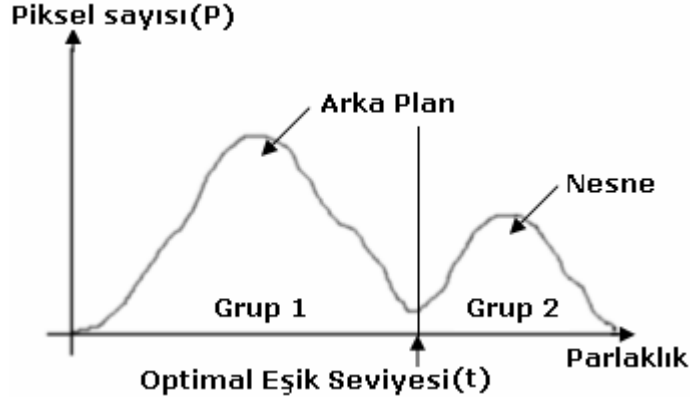
Bunun için çerçevelerin yazı bölgelerindeki pikseller karşılaştırılarak sahneler arası benzerlikler bulunur. Bunu yapmak için aşağıdaki adımlar uygulanır:

1. Benzerlikleri değerlendirilecek çerçevelere sadece değişmeyen yazı bölgesi kalacak şekilde maskeleme işlemi uygulanır.
2. Çerçeveler renkliden gri seviyesine dönüştürülür.
3. Otsu eşik seviyesi belirleme yöntemi [14] ile eşik seviyesi bulunur ve görüntü ikili resime dönüştürülür.
4. İkili resimler arasındaki Öklit Mesafesi hesaplanır.
5. Hesaplanan mesafe deneysel sonuçlar ile belirlenmiş eşik seviyesi ile karşılaştırılarak yazı bölgelerinin benzer olup olmadığına karar verilir.

### 2.3.2. Otsu Eşik Seviyesi Bulma Yöntemi

OTSU eşik seviye belirleme yöntemi gri seviyesi resimleri siyah-beyaz renklerden oluşan ikili resimlere dönüştürmek üzerine uygulanan bir yöntemdir. Bu yöntem, bimodel (Şekil 2.12) histogramlarda en optimum eşik seviyesi bulmayı hedeflemektedir. Bunun için

hedeflenen eşik seviyesi grup içindeki sapmayı minimum veya gruplar arasındaki sapmayı maksimum yapmalıdır. Gruplar, histogramda belirlenen eşik seviyesinin sol ve sağ tarafında kalan bölgelerdir.



Şekil 2.12: Bimodel renk parlaklık dağılımı

OTSU yönteminde, en büyük gri seviyesi değeri  $L$  ise  $0 \leq T \leq L-1$  olacak şekilde eşik seviyesi değiştirilerek, her eşik seviyesi için aşağıdaki işlemler tekrarlanır:

1. OTSU grup içi minimum sapmayı hedeflediği için ilk önce grupların olasılıklarını bulunur (Eşitlik 2.19). Burada  $t$  eşik seviyesini,  $P(i)$  ise  $i$ . Gri seviyesindeki piksel sayısını göstermektedir.

$$q_1(t) = \sum_{i=1}^t P(i) \quad q_2(t) = \sum_{i=t+1}^L P(i) \quad (2.19)$$

2. Daha sonra grupların ortalamaları bulunur (Eşitlik 2.20).

$$\mu_1(t) = \sum_{i=1}^t \frac{iP(i)}{q_1(t)} \quad \mu_2(t) = \sum_{i=t+1}^L \frac{iP(i)}{q_2(t)} \quad (2.20)$$

3. Grup içi sapmalar hesaplanır. (Eşitlik 2.21)

$$\sigma_1^2(t) = \sum_{i=1}^t [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \sigma_2^2(t) = \sum_{i=t+1}^L [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \quad (2.21)$$

4. Daha sonra grup için sapma hesaplanır (Eşitlik 2.22). Grup içi sapma, grubun içinde olan parlaklık olasılık değerlerine göre hesaplanmış standart sapmadır.

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \quad (2.22)$$

Sonuçta en az sapmayı sağlayan  $T$  değeri eşik seviyesi olarak kabul edilir. Toplam sapma değeri grup içi sapma ile gruplar arası sapma değerlerini toplamına eşittir (Eşitlik 2.23).

$$\sigma^2 = \sigma_w^2(t) + \sigma_B^2(t) \quad (2.23)$$

Toplam sapma değeri T değerine bağlı olmadan her zaman sabit bir değer olacağı için grup içi sapmada minimum olan T değeri aynı zamanda gruplar arası sapmanın da maksimum olduğu değer olacaktır. Bu yüzden gruplar arası maksimum sapmayı sağlayan T değeri aynı sonucu vermektedir. Bu yöntem ile T değeri rekürsiv şekilde bulunur.

1. Önce  $q_1$  ve  $m_1$  değerleri ilklendirilir (Eşitlik 2.24).

$$q_1(1) = P(1); \mu_1(0) = 0 \quad (2.24)$$

2. Daha sonra her  $t+1$  için gruplar arası sapma tekrar hesaplanır ve en büyük sapmayı veren T değeri eşik seviyesi olarak tanımlanır. Bunu yapmak için (Eşitlik 2.25) de grup olasılığı hesaplanır.

$$q_1(t+1) = q_1(t) + P(t+1) \quad (2.25)$$

3. Her iki grup için ortalama değerler bulunur (Eşitlik 2.26 ve 2.27).

$$\mu_1(t+1) = \frac{q_1(t)\mu_1(t) + (t+1)P(t+1)}{q_1(t+1)} \quad (2.26)$$

$$\mu_2(t+1) = \frac{\mu - q_1(t+1)\mu_1(t+1)}{1 - q_1(t+1)} \quad (2.27)$$

4. Bu değerler bulunduktan sonra gruplar arası sapma değeri Eşitlik 2.28-e göre hesaplanır.

$$\sigma_B^2(t) = q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2 \quad (2.28)$$

Eşitlik 2.28'de hesaplanan gruplar arası sapma değeri iki çerçeve arasında sahne geçişi olup olmadığını anlamak için kullanılmaktadır. Bu değer deneysel sonuçlar ile belirlenmiş eşik seviyesinden fazla ise iki çerçeve arası bir sahne geçişi olarak işaretlenir.

### 3. ÖN HAZIRLIK

Bu bölümde tasarlanan sistemin uygulamasında kullanılacak haber videosu veritabanının hazırlanması, farklı katmanlarda kullanılacak eğitim verisinin oluşturulması ve ihtiyaç duyulacak parametrelerin belirlenmesi işlemleri hakkında bilgi verilecektir.

#### 3.1. Haber Videosu Veritabanının Oluşturulması

Geliştirilen sistemin performansının ölçülmesi için haber programları farklı formatlarda olan 3 kanaldan farklı zamanlarda 8 adet haber programı kaydedilerek bir haber programı veritabanı oluşturulmuştur. Bir haber videosu ortalama 45 dakika sürmektedir. Eğer video frekansı (FPS) 25 çerçeve/saniye ise toplamda tahmini 67500 çerçeve oluşturulur (Eşitlik 3.1).

$$\text{Çerçeve sayısı} = 45 * 60 * 25 = 67500 \text{ (3.1)}$$

#### 3.2. Renk ve Kenar Değişimi Özellikleri Katmanı İçin Ön Hazırlık

Renk ve kenar özelliklerinin değişimine göre ardışık çerçevelerin benzerliğini ölçen birinci katmanda benzerlik kararının verilebilmesi için bazı eşik seviyelerinin belirlenmesi gerekir.

İlk katmanda ihtiyaç duyulan eşik seviyelerini belirlemek için rasgele seçilen 50 tane sahne geçişi olmayan ve 50 tane de sahne geçişi olan çerçeve eğitim verisi olarak seçilmiştir. Şekil 3.1’de sahne geçişi olmayan ardışık örnek çerçeveler, Şekil 3.2’de sahne geçişi olan ardışık örnek çerçeveler görülmektedir.



Şekil 3.1: Sahne geçişi olmayan örnek çerçeveler



Şekil 3.2: Sahne geçişi olan örnek çerçeveler

### 3.2.1. Renk Değişimi Analizi İçin Eşik Seviyesi Belirlenmesi

Renk değişimi analizinde ardışık çerçevelerin renk histogramlarının farklılıklarına bakılır. İlk olarak sahne geçişi olmayan 50 örnek çerçeve için mesafe hesabı yapılmıştır. Elde edilen sonuç Çizelge 3.1-de gösterilmektedir:

| Mesafe | Örnek sayısı |
|--------|--------------|
| <100   | 3            |
| <200   | 36           |
| <300   | 3            |
| <400   | 4            |
| <500   | 1            |
| <600   | 2            |
| <700   | 0            |
| <800   | 1            |

**Çizelge 3.1: Sahne geçişi olmayan örneklerde ardışık çerçevelerde mesafe**

Daha sonra sahne geçişi olan 50 örnek ardışık çerçeve için mesafe hesabı yapılmıştır. Bu çerçeveler için elde edilen sonuçlar Çizelge 3.2’de görülmektedir.

| Mesafe | Örnek sayısı |
|--------|--------------|
| <1000  | 4            |
| <1500  | 12           |
| <2000  | 9            |
| <3000  | 17           |
| <4000  | 5            |
| <7500  | 2            |
| <10000 | 1            |

**Çizelge 3.2: Sahne geçişi olan örneklerde ardışık çerçevelerde mesafe**

Bu hesaplamalar sonucunda renk değişim farkı için eşik seviyesi Çizelge 3.1 ve Çizelge 3.2’deki değerler karşılaştırılarak ortalama 800 olarak belirlenmiştir. Şekil 3.3’de sahne geçişi olmayan örnek ardışık çerçeveler ve gri seviyesindeki histogramları görülmektedir.

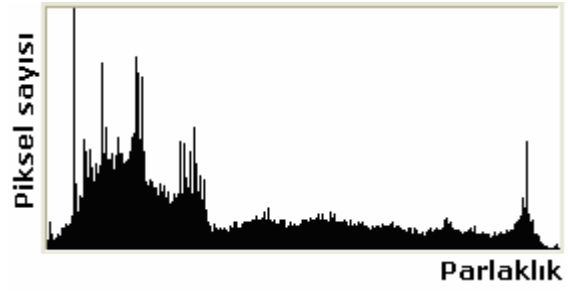
Şekil 3.3a ve Şekil 3.3c’deki iki çerçevenin histogramlarının mesafesi 279.85 olarak hesaplanmıştır. Mesafe hesabı eşik seviyesinden küçük olduğu için bu iki çerçeve aynı sahneye ait olduğu görülmektedir.

Şekil 3.4’de sahne geçişi olan örnek ardışık çerçeveler ve gri seviyesindeki histogramları görülmektedir.

Şekil 3.4a ve Şekil 3.4c’deki iki çerçevenin histogramlarının meafesi 4793.066 olarak hesaplanmıştır. Mesafe hesabı eşik seviyesinden büyük olduğu için burada bir sahne geçişi olduğu kabul edilmiştir.



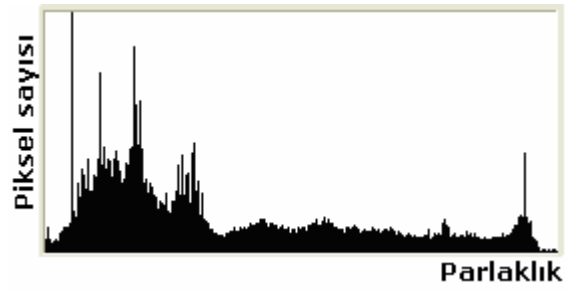
(a)



(b)



(c)

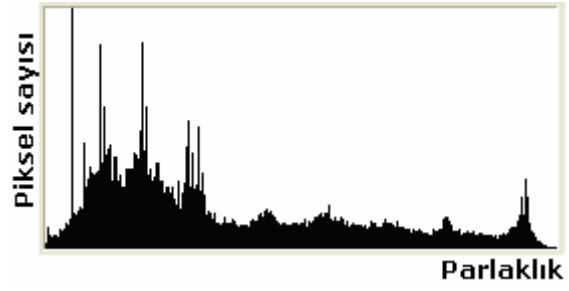


(d)

Şekil 3.3: a-c) Sahne geçişi olmayan çerçeveler. b-d) Çerçevelere ait gri seviyesi histogramları



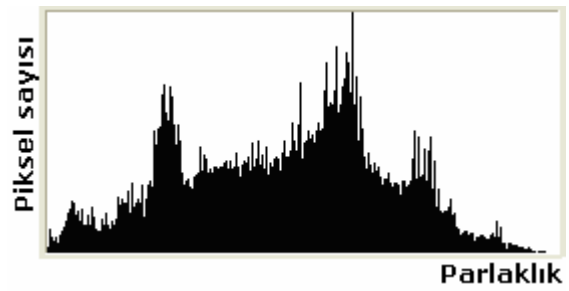
(a)



(b)



(c)



(d)

Şekil 3.4: a-c) Sahne geçişi olan çerçeveler. b-d) Çerçevelere ait gri seviyesi histogramları

### 3.2.2. Kenar Değişimi Analizi İçin Eşik Seviyesi Belirlenmesi

Renk değişimi analizinde farklı olduklarına karar verilen çerçevelerin gerçekten farklı olup olmadıklarını anlamak için kenar değişimi farklılıkları değerlendirilir. Ölçülen kenar değişimi farklılıklarında kullanılacak eşik seviyesinin belirlenmesi gereklidir. için sahne geçişi olmayan 50 örnek çerçeveye işlem uygulandığında Çizelge 3.3'deki sonuç elde edilmiştir.

| Mesafe | Örnek sayısı |
|--------|--------------|
| <0.03  | 5            |
| <0.06  | 13           |
| <0.09  | 14           |
| <0.12  | 7            |
| <0.15  | 8            |
| <0.18  | 3            |

Çizelge 3.3: Sahne geçişi olmayan örneklerde kenar piksel değişimi mesafe dağılımı

Aynı işlem sahne geçişi olan 50 örnek çerçeveye uygulandığında Çizelge 3.4'deki sonuçlar elde edilmiştir.

| Mesafe | Örnek sayısı |
|--------|--------------|
| <0.30  | 2            |
| <0.40  | 4            |
| <0.45  | 11           |
| <0.50  | 18           |
| <0.60  | 11           |
| <0.70  | 4            |

Çizelge 3.4: Sahne geçişi olan örneklerde kenar piksel değişimi mesafe dağılımı

Bu sonuçlar incelenerek kenar pikseli farkı için eşik seviyesi Çizelge 3.3 ve 3.4'deki değerler karşılaştırılarak ortalama **0.2** olarak belirlenmiştir.

Şekil 3.5-3.8 arasında sahne geçişi olmayan ardışık çerçevelere kenar değişimi analizi işlemleri uygulanmasının adımları görülmektedir.

Şekil 3.5'deki her iki çerçeveye kenar pikselleri belirleme işlemi uygulanır. Bunun için Canny kenar belirleme yöntemi kullanılır (Şekil 3.6).



Şekil 3.5: Sahne geçişi olmayan ardışık spiker çerçeveleri



Kenar piksel toplamı: 6157

(a)



Kenar piksel toplamı: 6204

(b)

Şekil 3.6: Çerçevelere kenar belirleme yöntemi uygulanmış resimler

Daha sonra bu resimlere genişletme ve ters çevirme işlemlerini uygulanır(Şekil 3.7).



(a)



(b)

Şekil 3.7: Çerçevelere genişletme ve ters çevirme işlemi uygulanmış resimler

Oluşan yeni resimler ve (Şekil 3.7a ve Şekil 3.7b) kenar pikselleri belirlenen resimlerle (Şekil 3.6a ve Şekil 3.6b) çapraz şekilde VE işlemi uygulanır (Şekil 3.8). Bu işlemin sonucunda oluşan resimlerde (Şekil 3.8a ve Şekil 3.8b) tekrar kenar pikselleri sayılır.



Kenar piksel toplamı: 508



Kenar piksel toplamı: 613

**Şekil 3.8: Çerçevelere VE işlemi uygulanması sonucu oluşan resimler**

Son olarak yeni piksel sayılar ile eski kenar pikseller sayılarının oranını her iki resim için hesaplanır ve büyük olanını alınır (Eşitlik 3.5).

$$\text{Kenar Piksel Değişim Oranı}_N = 508 / 6157 = 0.0825$$

$$\text{Kenar Piksel Değişim Oranı}_{N+1} = 613 / 6204 = \mathbf{0.0988 \text{ (3.5)}}$$

Bulunan bu değer iki çerçeve arasındaki kenar değişimi oranını göstermektedir. Bu değer belirlenen eşik seviyesi olan 0.20 değerinden küçük olduğu için sahne geçişi olmadığı ve bu çerçevelerin aynı sahneye ait olduğu varsayılır.

Bir sonraki örnekte Şekil 3.9-3.12 arasında sahne geçişi olan çerçeveler üzerinde yapılan işlemler görülmektedir.

**Şekil 3.9: Sahne geçişi olan örnek ardışık çerçeveler**

Şekil 3.9'de görülen ardışık iki çerçeveye Canny kenar belirleme yöntemi uygulanır. (Şekil 3.10)



Kenar piksel toplamı: 6987

(a)



Kenar piksel toplamı: 10417

(b)

**Şekil 3.10: Çerçevelere kenar belirleme yöntemi uygulanmış resimler**

Genişletme ve ters çevirme işlemlerini uygulanır(Şekil 3.11).



(a)



(b)

**Şekil 3.11: Çerçevelere genişletme ve ters çevirme işlemi uygulanmış resimler**

Son olarak da çapraz VE işlemi uygulanır (Şekil 3.12).



Kenar piksellerin toplamı: 1998



Kenar piksellerin toplamı: 5319

**Şekil 3.12: Çerçevelere VE işlemi uygulanması sonucu oluşan resimler**

Son olarak yeni piksel sayılar ile eski kenar pikseller sayılarının oranını her iki resim için hesaplanır ve büyük olan oran alınır (Eşitlik 3.6).

$$\text{Kenar Piksel Değişim Oranı}_N = 1998 / 6987 = 0.2859$$

$$\text{Kenar Piksel Değişim Oranı}_{N+1} = 5319 / 10417 = \mathbf{0.5106 (3.6)}$$

Büyük olan değer **0.5106** belirlenen 0.20 eşik seviyesinden büyük olduğu için bu iki çerçeve arası sahne geçişi olduğu kararı verilir.

### 3.3 Spiker Tanıma Katmanı İçin Ön Hazırlık

Bu katmanda yapılan yüz tanıma işlemi için PCA yöntemi kullanılmıştır. Bu yöntemde, sistemin önce tanınması istenen kişilere ait mevcut yüz resimleri ile eğitilmesi gerekir.

#### 3.2.3. Yüz Tanıma Sisteminin Eğitilmesi

Spiker yüzlerine ait yüz resmi veritabanını oluşturmak için haber videosu veritabanındaki video görüntülerinden spiker yüzleri olan çerçeveler seçilmiştir. Mevcut 67500 çerçeveden spiker yüzü olan yaklaşık 10000 çerçeve belirlenmiştir. Daha sonra OpenCV kütüphanesi kullanılarak belirlenmiş çerçevelerde yüz bulma işlemi gerçekleştirilerek bulunan yüzler ayrı bir dizine kopyalanmıştır. Burada arka arkaya gelen çerçeveler de olduğu için birbirine çok benzeyen yüzler de eğitim verisine eklenmiş olacaktır. Eğitim setinin daha küçük ve daha farklı durumları içeren yüzler içermesi için yüz bulma işlemi her çerçeve için değil, her 10 çerçevede bir uygulanmıştır. Eğitim seti için 5 haber spikerine ait toplam 5000 tane yüz bulunmuştur.

Yüz tanımda PCA uygulanacağı için 5000 yüz üzerinde işlem yapmak çok zaman alacaktır. Bu yüzden daha küçük ve daha çok başarı sağlayacak bir eğitim setinin oluşturulması gerekir. En uygun yüz resimlerini seçmek için k-fold çapraz geçerlilik sınaması yöntemi [15] ile seçim yapılmıştır. Bunu yapmak için bütün yüzler bir arada toplandıktan sonra belirli K değerleri için bu resimlerden K tanesi eğitim için ayrılmış geri kalan resimler ise test için kullanılmıştır. Her adımda hata oranı hesaplanmış, K adım sonunda hataların ortalaması alınarak ortalama hata bulunmuştur. Bu işlem tamamlandığında hangi K değerinin daha başarılı sonuç verdiği ortaya çıkar. Aynı K değeri için bu işlem farklı özvektör sayısı kullanılarak yapılmıştır. Bu bilgiler Çizelge 3.5, 3.6 ve 3.7-de görülmektedir.

**K=128 Öz vektör Sayısı = 64**

| K | Süre (ms) | Ortalama Mesafe |
|---|-----------|-----------------|
| 0 | 21859     | 26.26539459     |
| 1 | 14578     | 25.51623684     |
| 2 | 14672     | 26.70082282     |
| 3 | 14735     | 32.19648591     |
| 4 | 14609     | 27.50009754     |
| 5 | 14484     | 31.86256754     |
| 6 | 14594     | 30.60362475     |
| 7 | 14484     | 35.75902218     |
| 8 | 14516     | 27.76493446     |
| 9 | 14422     | 22.81722914     |

**Çizelge 3.5: K değerlerine göre mesafe hesabı 1**

En küçük mesafe: 22.817229142576515

En başarılı K Değeri: 9

**K=128 Öz vektör Sayısı = 32**

| K | Süre (ms) | Ortalama Mesafe |
|---|-----------|-----------------|
| 0 | 9921      | 19.07295147     |
| 1 | 9875      | 20.34045871     |
| 2 | 9782      | 19.02140233     |
| 3 | 9656      | 21.25855561     |
| 4 | 9562      | 19.51970119     |
| 5 | 9610      | 22.47497463     |
| 6 | 9656      | 23.47926082     |
| 7 | 9594      | 27.36380878     |
| 8 | 9609      | 19.22539305     |
| 9 | 9750      | 16.3990473      |

**Çizelge 3.6: K değerlerine göre mesafe hesabı 2**

En küçük mesafe: 16.399047299257738

En başarılı K Değeri: 9

**K=128 Öz vektör Sayısı = 16**

| K | Süre (ms) | Ortalama Mesafe |
|---|-----------|-----------------|
| 0 | 6844      | 11.50996757     |
| 1 | 6828      | 12.42916403     |
| 2 | 6844      | 15.21695154     |
| 3 | 6813      | 14.80270029     |
| 4 | 6828      | 14.42267613     |
| 5 | 6875      | 16.13768151     |
| 6 | 6844      | 16.47080376     |
| 7 | 6875      | 18.54861026     |
| 8 | 6859      | 13.65412224     |
| 9 | 6969      | 10.50189877     |

**Çizelge 3.7: K değerlerine göre mesafe hesabı 3**

En küçük mesafe: 10.501898770493789

En başarılı K Değeri: 9

Denemeler sonucunda bütün özdeğerler için en iyi sonucun K değerinin 9 olduğu durum için sağlandığı gözlemlenmiştir. Bu durumda yüz tanıma için kullanılacak olan eğitim verisi en küçük mesafe değerinin olduğu yüzler olarak belirlenmiştir. Çizelgelarda ortalama mesafe sonuçları eğitim verisi ile validasyon için kullanılan spiker yüzleri arasındaki PCA işlemi sonucunda oluşan mesafe değerleridir.

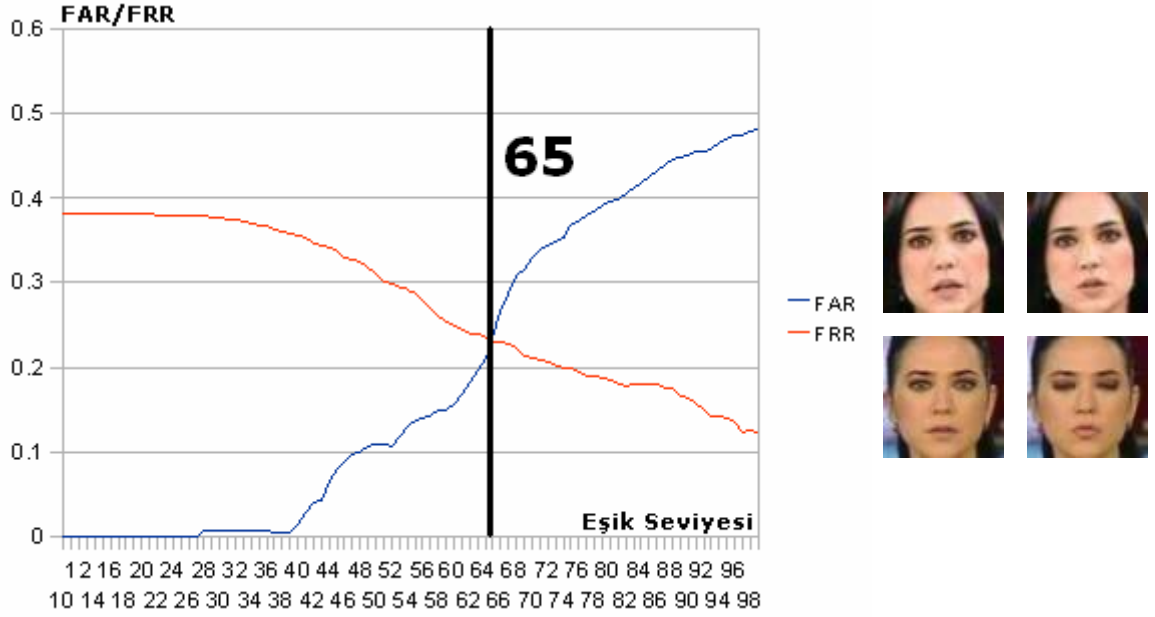
Buradaki yüz tanıma işlemi sınıflandırma değil, tanımadır. Dolayısıyla bulunan yüzün spiker yüzü olup olmadığına karar verilmesi gerekir. Eğitilmiş her spiker yüzü için tanımda kullanılacak eşik seviyesi belirlenir.

### 3.2.4. Eşik Seviyelerinin Belirlenmesi

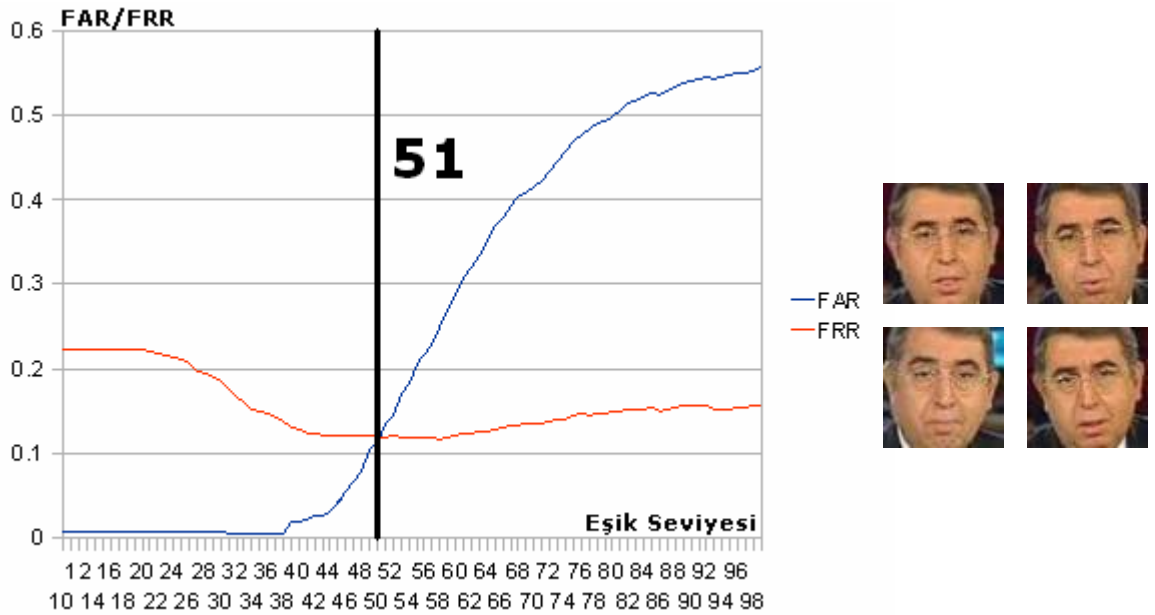
Eşik seviyelerinin belirlenmesi için test verisi kullanılarak Yanlış Kabul Oranı( FAR -False Acceptance Rate) ve Yanlış Red Oranı( FRR - False Rejection Rate) hesaplanır. FAR spiker yüzü olmadığı halde spiker olarak tanınan yüzlerin oranı ve FRR de spiker yüzü olduğu halde tanınamayan yüzlerin oranıdır. FAR oranı ne kadar düşük olursa yanlış tanıma sayısı o kadar azalacaktır. FRR ne kadar düşük olursa spiker o kadar çok tanınmış olacaktır. Eşik seviyesi belirlerken her bir T eşik seviyesi değeri için test verisi ile tanıma işlemi yapılarak FAR ve FRR oranları hesaplanır. Daha sonra sistemin ihtiyacına göre FAR ile FRR'ın kesiştiği yerde veya duruma göre yanlış tanınmama önemli ise FAR'ın sifıra yakın olduğu yerde veya doğru tanıma önemli ise FRR'nin sifıra yakın olduğu yerde bir eşik seviyesi belirlenmesi gerekmektedir.

Bu çalışmada FAR değerinin sifıra yakın olması daha önemlidir. Çünkü sistem spiker yüzü bulduğu zaman o çerçeveden sonraki çerçeveler eğer yüzün bulunduğu çerçeveye renk özellikleri bakımından benziyor ise bunlar da bir spiker çerçevesi olarak kabul edilir. Böylece her çerçeve için uzun süren yüz bulma ve bulduktan sonra o yüzün spiker olup olmadığına bakma işlemlerinden kurtulmuş olunur. Fakat bu tasarımın kötü tarafı ise spiker olmayan bir yüzün spiker olarak tanındığı bir durumda, yani FA, durumunda o çerçeveye benzeyen bütün çerçevelerin de spiker çerçevesi olarak işaretlenmesi sonucuna varılır. Bu yüzden yanlış tanımanın en az seviyede olması gerekir. Bu yüzden eşik seviyesini FAR'ın sifıra yakın olduğu yerden seçilmesi gerekir. Bu durumda aslında FRR'in yüksek çıkması kabul edilmiş olunur. Bu nedenle sistem eşik seviyesi daha küçük belirlendiği için spiker olduğu halde bir kısım yüzleri tanıyamayacaktır.

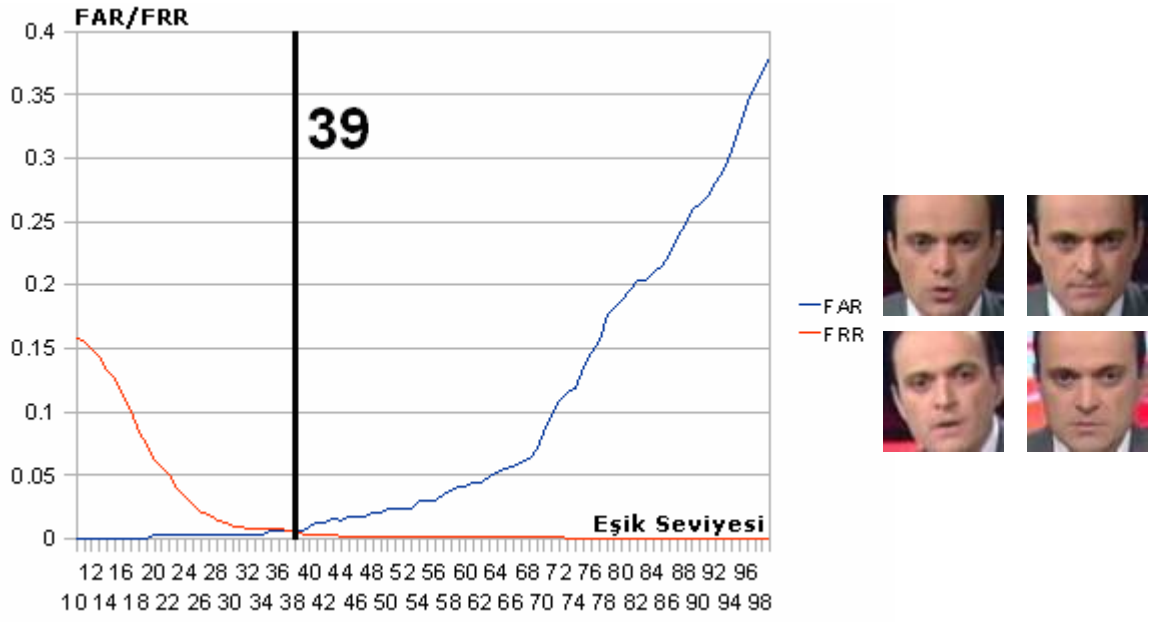
Sistem bir kere spiker yüzünü tanıdığı bir çerçeve bulduğu zaman geriye ve ileriye doğru çerçeveleri tarar ve renk olarak benzeyen bütün çerçeveleri spiker çerçevesi olarak etiketlediği için hem başarılı bir spiker tanıma işlemi hem de hızlı çalışan bir sistem tasarlanmış olur. Şekil 3.13, 3.14, 3.15, 3.16 ve 3.17’de farklı spikerler için FAR ve FRR değişimleri görülmektedir. Burada her spiker için farklı eşik seviyesi kullanılmaktadır. Eşik seviyeleri FAR ile FRR’in kesiştiği noktalar olarak belirlenmiştir.



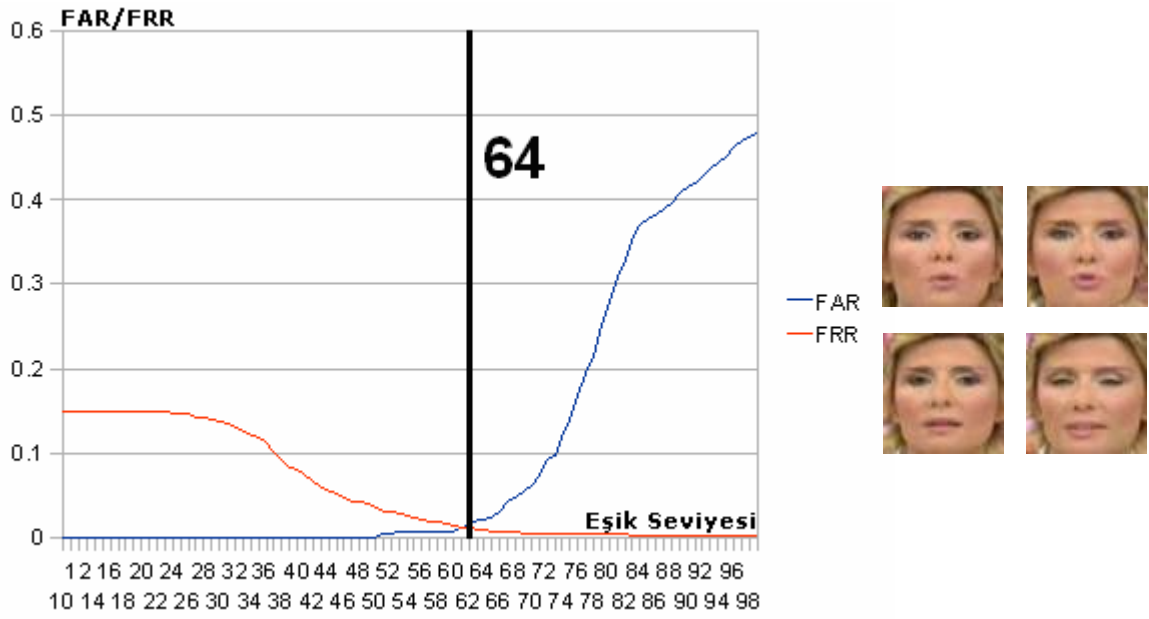
Şekil 3.13: Sağdaki spiker için FAR ve FRR grafiği



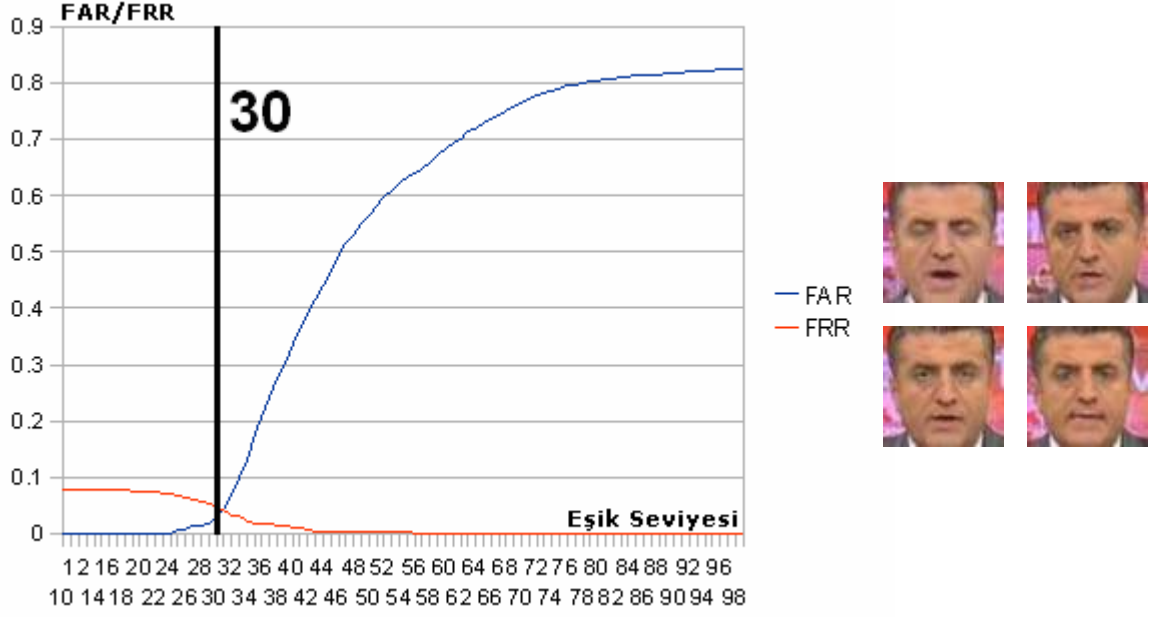
Şekil 3.14: Sağdaki spiker için FAR ve FRR grafiği



Şekil 3.15: Sağdaki spiker için FAR ve FRR grafiği



Şekil 3.16: Sağdaki spiker için FAR ve FRR grafiği



Şekil 3.17: Sağdaki spiker için FAR ve FRR grafiği

Spiker tanıma katmanı sonucunda oluşan sahneler iki sınıfa ayrılır.

- S: Spikerin konuştuğu sahneler
- V: Haberle ilgili video görüntüler (spikerin olmadığı sahneler)

Bu katman son olarak arka arkaya gelen aynı sınıftaki sahneleri birleştirilerek tek bir sahne haline getirir. Örneğin arka arkaya gelen beş tane V sahnesi var ise, onların hepsi tek bir V sahnesi olarak birleştirilir.

Böylece her bir haberi bir spiker konuşması ve bir de haber videosu olarak düşündüğümüzde her S-V çifti bir haber olarak algılanır. Fakat bir haber ile ilgili sahneler birden fazla S-V çiftleri halinde olabilir. Bu durum yazı tanıma katmanında analiz edilir.

### 3.3. Yazı Tanıma Katmanı İçin Öz Hazırlık

Bu katmanda önce maskeleyme işlemi uygulanarak, çerçevedeki yazı bölgesi arka plandan ayrılır. Daha sonra yazı bölgesi renkliden gri seviyesine dönüştürülür. Daha sonra OTSU eşik seviyesi belirleme yöntemi ile bu resim siyah-beyaz hale getirilir. Son olarak da iki çerçeve arası yazı bölgeleri karşılaştırılarak aradaki farka bakılır.

Bu katman için yapılması gereken ön işlemler şunlardır :

1. Yapılan değerlendirmelerde farklı haber programları için yazı bölgelerinin yerinin sabit olduğu görülmüştür. Bu nedenle her kanalın haber programı için örnek görüntüler değerlendirilerek yazı bölgesinin yeri belirlenmelidir.
2. Yazı bölgesi arka plandan ayrıldıktan sonra önce gri seviyesine, sonra da siyah-beyaza yani ikili seviyeye dönüştürülür. Siyah-beyaz dönüşümü sırasında kullanılacak eşik seviyesi değeri örnek yazı bölgesi resimleri üzerinde denenerek belirlenmiştir.
3. Çerçevelerin yazı bölgelerinin mesafelerine göre değerlendirilerek aynı olup olmadığının karşılaştırılması için eşik seviyesi yine örnek resimler kullanılarak belirlenmiştir.

Bu işlemler aşağıdaki örnek resimlerde uygulamalı olark gösterilmektedir. Önce karşılaştırılacak çerçeveler seçilir (Şekil 3.18).



Şekil 3.18: Yazı analizi için örnek çerçeveler

Daha sonra yazı bölgeleri maskelenerek arka plandan ayrılır ve gri tona dönüştürülür (Şekil 3.19).



Şekil 3.19: Yazı bölgeleri

OTSU yöntemi sonucu siyah-beyaz resimler elde edilir (Şekil 3.20).



Şekil 3.20: OTSU sonucu elde edilen siyah-beyaz resimler

Son olarak bu iki resim arasında mesafe farkı hesaplanarak iki sahnenin aynı habere ait olup olmadığına bakılır.

### 3.4. Sistem Aşamalarının Özeti

Sistemin çalışma aşamaları özet olarak aşağıdaki gibidir :

1. Değerlendirilmesi istenen kaynak haber videosu çerçevelere ayrılır.
2. Daha sonra ilk analiz katmanı olan renk ve kenar pikselleri değişimine göre analiz için gereken eşik seviyeleri bütün çerçeveler üzerinde uygulanır. Ardışık çerçevelerden benzer olarak işaretlenenler ayrı gruplar oluşturacak şekilde, küçük boyutlarda bir çok çekimler elde edilir. Bu grupları göstermek için bir dosyada sadece her parçanın başlangıç ve bitiş çerçeve bilgisi olarak saklanır.
3. Yüz tanıma ile spiker tanıma işlemi yapılır. Birinci aşama sonucunda elde edilen her grup üzerinde ayrı ayrı analiz yapılır. Bu aşamada bir gruptaki herhangi bir çerçevede spikerin yüzü bulunduktan sonra yüz tanıma işlemi durdurulur. Bakılan bölüm spikere ait bölüm olarak (S) işaretlenir. Eğer bakılan bölümde hiçbir çerçevede spiker yüzü bulunamazsa bu bölüm habere ait olarak (V) işaretlenir.
4. Önceki adımlarda yapılan işlemler sonunda haber geçişlerinin bulunamaması durumunda, benzerlikler sahnelerin yazı bölgesi olarak işaretlenen bölgedeki özelliklerine bakılarak bulunmaya çalışılır. Yazı olarak benzer bulunan sahneler birleştirilerek tek bir sahne haline getirilir. Bu katmanın sonucu olarak haber programını haber geçişlerine göre bölütlenme tamamlanmış bulunur.

#### 4. UYGULAMA

Uygulama bölümünde sistemin başarısından ve başarıyı etkileyen faktörlerden bahsedilmiştir. Eğitim ve test için üç farklı yerel televizyon kanalı haberleri kullanılmıştır. Değişik kanalların seçilmesinin sebebi, haber programı yapılarının birbirinden farklı olması ve karışıklık seviyelerinin farklılık göstermesidir. Haber videoları, ortalama 45 dakika süren içeriği zengin olan video görüntüleridir. Haberlerle ilgili görüntülü ve sesli çok değişken bilgi içermektedir.

Türk kanallarındaki haber programları genel olarak ilk önce spiker konuşması ve onu takip eden haber ile ilgili videolardan oluşmaktadır. Dolayısıyla haber spikeri görüldükten sonra spikerin bir sonra görüleceği çerçeveye kadar olan görüntüler aynı habere ait olarak yorumlanabilir. Fakat bu durum her zaman geçerli değildir. Bazen spikerden sonra bir telefon konuşması veya spiker konuştuğundan sonra haberle ilgili video gösterilmeden başka bir habere geçilmesi gibi durumlar oluşabilir. Sistem katmanlı yapıda tasarlandığı için, önceki örnekteki gibi daha karmaşık yapıların algılanması gerektiğinde, farklı katmanlar tasarlanarak sisteme eklenebilir.

Aşağıdaki bölümlerde her televizyon kanalı için yapılan testlerden ve sistemin başarısından bahsedilmektedir. Ayrıca sistemin yetersiz olduğu durumlara değinilerek alternatif çözümlerin nasıl olabileceği tartışılmıştır. Her örnek için videoların Çizelge 1’de belirtilen özellikleri için Çizelge 4.2-deki değerler çıkarılmış ve bu değerler üzerinden başarı oranı Eşitlik 4.3-e göre hesaplanmıştır.

|                                 |                                         |
|---------------------------------|-----------------------------------------|
| <b>Toplam Süre</b>              | Haber programının toplam süresi         |
| <b>Dosya Boyutu</b>             | Haber programı dosyasının boyutu        |
| <b>Toplam Çerçeve Sayısı</b>    | Video görüntüdeki toplam çerçeve sayısı |
| <b>Çerçeve Frekansı (FPS)</b>   | Çerçeve frekansı                        |
| <b>Resim Boyutları (piksel)</b> | Çerçevelerin boyut özellikleri          |
| <b>Video Tipi</b>               | Video görüntü dosya tipi                |

**Çizelge 4.1: Örnek video özellikleri ve açıklamaları**

|                                       |                                                                |
|---------------------------------------|----------------------------------------------------------------|
| <b>İşlenen Çerçeve Sayısı</b>         | Sistem tarafından işlenen çerçeve sayısı                       |
| <b>1. Katman Haber Sayısı</b>         | Renk ve Kenar değişimi katmanının bulunduğu haber geçiş sayısı |
| <b>2. Katman Haber Sayısı</b>         | Spiker tanıma katmanının bulunduğu haber geçiş sayısı          |
| <b>3. Katman Haber Sayısı</b>         | Yazı bölgesi analizi katmanının bulunduğu haber geçiş sayısı   |
| <b>Gerçek Haber Sayısı</b>            | Haber programının gerçekte olan haber sayısı                   |
| <b>Doğru Bulunan Haber Geçişleri</b>  | Doğru bulunan haber geçiş sayısı                               |
| <b>Yanlış Bulunan Haber Geçişleri</b> | Yanlış bulunan haber geçiş sayısı                              |
| <b>Doğru Kabul Oranı</b>              | Doğru haber geçişlerinin bulunma oranı (Eşitlik 4.1)           |
| <b>Yanlış Kabul Oranı</b>             | Yanlış haber geçişlerinin bulunma oranı (Eşitlik 4.2)          |
| <b>Başarı Oranı (%)</b>               | Sistemin başarı oranı (Eşitlik 4.3)                            |

**Çizelge 4.2: Analiz sonucu çıkan değerler ve açıklamaları**

Doğru kabul oranını doğru bulunan haber geçişlerinin gerçek haber sayısına olan oranıdır (Eşitlik 4.1). Yanlış kabul oranı ise yanlış bulunan haber geçişlerinin toplam işlenen çerçeve sayısına olan oranıdır (Eşitlik 4.2). Çünkü analiz başlamadan önce haber programı birbirinden bağımsız binlerce çerçeveden oluşan bir görüntüdür. Her çerçeve bir haber geçişi olabilir. Katmanlarda yapılan analizler sonucunda benzer çerçeveler elendiğinden bu değer çok küçülür. Sistemin genel başarısı Eşitlik 4.3-te bulunan iki oran kullanılarak hesaplanır.

$$\text{Doğru Kabul Oranı (DKO)} = \frac{\text{Doğru bulunan gecisler}}{\text{Gerçek haber sayisi}} \quad (4.1)$$

$$\text{Yanlış Kabul Oranı (YKO)} = \frac{\text{Yanlis bulunan gecisler}}{\text{Toplam cerceve sayisi}} \quad (4.2)$$

$$\text{Sistem Başarı Oranı} = \frac{DKO}{DKO + YKO} \cdot 100 \quad (4.3)$$

#### 4.1. Farklı Kanallar İçin Haber Programları ile Testler

Bu çalışmada 3 farklı kanalın haber programları değerlendirilmiştir. Kanallar seçilirken haber programı formatlarının birbirinden farklı olması kriter alınmıştır. Bu kanallardan alınan görüntüler üzerinde yapılan test işlemlerinin sonuçları kanal isimleri verilmeden, kanallar numara ile belirtilerek değerlendirilecektir.

#### 4.1.1. Kanal-1 Haber Programı İçin Değerlendirme

Bu kanalda haber programı çok düzenli bir yapıya sahip olduğu için bölütleme işlemi çok başarılı sonuçlar vermiştir. Haberlerin tamamı spiker konuşması ve onu takip eden haberle ilgili görüntülerden ibarettir. Ayrıca haber spikerinin fazla hareket etmeden, devamlı ekrana dik bakması spiker tanıma başarısını artıran önemli bir etken olmuştur. Şekil 4.1’de bu kanaldan alınan örnek haber çerçeveleri görülmektedir.



Şekil 4.1: Kanal-1 haberi örnek çerçeveleri

|                                 |                     |
|---------------------------------|---------------------|
| <b>Toplam Süre</b>              | 34 dakika 46 saniye |
| <b>Dosya Boyutu</b>             | 245 Mb              |
| <b>Toplam Çerçeve Sayısı</b>    | 52346               |
| <b>Çerçeve Frekansı (FPS)</b>   | 25                  |
| <b>Resim Boyutları (piksel)</b> | 384x288             |
| <b>Video Tipi</b>               | MPEG-4              |

Çizelge 4.3: Kanal-1 örnek video özellikleri

|                                       |                 |
|---------------------------------------|-----------------|
| <b>İşlenen Çerçeve Sayısı</b>         | 52346           |
| <b>1. Katman Haber Sayısı</b>         | 915             |
| <b>2. Katman Haber Sayısı</b>         | 15              |
| <b>Gerçek Haber Sayısı</b>            | 15              |
| <b>Doğru Bulunan Haber Geçişleri</b>  | 15              |
| <b>Yanlış Bulunan Haber Geçişleri</b> | 0               |
| <b>Doğru Kabul Oranı</b>              | $15 / 15 = 1$   |
| <b>Yanlış Kabul Oranı</b>             | $0 / 52346 = 0$ |
| <b>Başarı Oranı (%)</b>               | 100             |

Çizelge 4.4: Kanal-1 örneği analiz sonuçları

Çizelge 4.3 ve Çizelge 4.4’de Kanal-1 ile ilgili değerler görülmektedir. İlk katman sonucunda belirlenen 915 tane sahne ikinci katmanda spiker yüzünün değerlendirilmesi ile 15 sahneye indirgenmiştir. Bu sonuç gerçek verilerle karşılaştırıldığında başarı oranı 100%’dür. Bunun

sebebi Kanal-1 haberlerinin S-V çiftine uyumlu olması yani her haberin spiker konuşması ile başlaması ve onu takip eden video görüntüsü ile bitmesidir. Ayrıca buradaki başarıyı etkileyen diğer bir husus da spiker olan sahnelerde yüz tanıma işleminin başarı ile yapılmış olmasıdır.

#### 4.1.2. Kanal-2 Haber Programı İçin Değerlendirme

Kanal-2 haber programları düzen olarak tam bir standarda sahip değildir. Haberlere ayrılan süreler çok farklılık göstermektedir. Bazı haberler 2 dakika sürerken bazı haberler 20 dakika sürebilmektedir. Bunun sebebi gündem teşkil eden haberlere daha geniş yer ayrılmasıdır. Bu durumda sadece spiker konuşması değil haberle ilgili konukların programa katılması veya spikerin defalarca aynı konu üzerinde konuşması analizi zorlaştırmaktadır. Şekil 4.2’de Kanal-2’nin örnek çerçeveleri görülmektedir.



Şekil 4.2: Kanal-2 haber örnek çerçeveleri

|                          |                          |
|--------------------------|--------------------------|
| Toplam Süre              | 1 saat 3 dakika 3 saniye |
| Dosya Boyutu             | 455 Mb                   |
| Toplam Çerçeve Sayısı    | 116279                   |
| Çerçeve Frekansı (FPS)   | 29                       |
| Resim Boyutları (piksel) | 384x240                  |
| Video Tipi               | MPEG-4                   |

Çizelge 4.5: Kanal-2 örnek 1 video özellikleri

|                                       |                       |
|---------------------------------------|-----------------------|
| <b>İşlenen Çerçeve Sayısı</b>         | 60000                 |
| <b>1. Katman Haber Sayısı</b>         | 790                   |
| <b>2. Katman Haber Sayısı</b>         | 29                    |
| <b>3. Katman Haber Sayısı</b>         | 11                    |
| <b>Gerçek Haber Sayısı</b>            | 8                     |
| <b>Doğru Bulunan Haber Geçişleri</b>  | 8                     |
| <b>Yanlış Bulunan Haber Geçişleri</b> | 3                     |
| <b>Doğru Kabul Oranı</b>              | $8 / 8 = 1$           |
| <b>Yanlış Kabul Oranı</b>             | $3 / 60000 = 0.00005$ |
| <b>Başarı Oranı (%)</b>               | 99                    |

**Çizelge 4.6: Kanal-2 örneği 1 analiz sonuçları**

Çizelge 4.5 ve Çizelge 4.6’de Kanal-2 ile ilgili değerler görülmektedir. İlk katman sonunda elde edilen 790 tane sahne ikinci katmanda birleştirilerek 29 sahneye kadar azalmaktadır. Son katmanda haber geçişi sayısı 11 olarak bulunmuş ve analiz işlemi tamamlanır. Sonuçlar gerçek verilerle karşılaştırıldığında 99% başarılı sonuç elde edildiği görülmektedir. Haber geçişi olmayan çerçevelerin yanlış yakalanmasının sebebi haberlerin çok uzun sürmesi ve bir çok kez S-V çiftlerinin oluşmasıdır. Bu yüzden her oluşan çift bir habere karşılık olaark düşünüldüğü için olması gerekenden fazla haber bulunmuştur. Bu çiftlerin üçüncü katmanda büyük ölçüde azaldığı görülmektedir.

#### **4.1.3. Kanal-2 Haber Programı İçin Değerlendirme**

Önceki bölümde de belirtildiği gibi Kanal-2 haber programları düzen olarak daha karmaşık bir yapıya sahiptirler. Bunun dışında başarıyı etkileyen durum haberler arası gelecek olan haberlerle ilgili çerçevelerin yeralmasıdır. Bu video sahnelerini analiz katmanları haber video görüntüsü olarak işaretlenir. Fakat aslında bunlar sadece gelecek olan haberlere ait olan sahnelerdir. Şekil 4.3’de Kanal-2’nin örnek çerçeveleri görülmektedir.



Şekil 4.3: Kanal-2 haber örnek 2 çerçeveleri

|                                 |                           |
|---------------------------------|---------------------------|
| <b>Toplam Süre</b>              | 1 saat 24 dakika 6 saniye |
| <b>Dosya Boyutu</b>             | 603 Mb                    |
| <b>Toplam Çerçeve Sayısı</b>    | 128924                    |
| <b>Çerçeve Frekansı (FPS)</b>   | 25                        |
| <b>Resim Boyutları (piksel)</b> | 384x288                   |
| <b>Video Tipi</b>               | MPEG-4                    |

Çizelge 4.7: Kanal-2 örnek 2 video özellikleri

|                                       |                       |
|---------------------------------------|-----------------------|
| <b>İşlenen Çerçeve Sayısı</b>         | 60000                 |
| <b>1. Katman Haber Sayısı</b>         | 1081                  |
| <b>2. Katman Haber Sayısı</b>         | 14                    |
| <b>3. Katman Haber Sayısı</b>         | 7                     |
| <b>Gerçek Haber Sayısı</b>            | 5                     |
| <b>Doğru Bulunan Haber Geçişleri</b>  | 5                     |
| <b>Yanlış Bulunan Haber Geçişleri</b> | 2                     |
| <b>Doğru Kabul Oranı</b>              | $5 / 5 = 1$           |
| <b>Yanlış Kabul Oranı</b>             | $2 / 60000 = 0.00003$ |
| <b>Başarı Oranı (%)</b>               | 99                    |

Çizelge 4.8: Kanal-2 örneği 2 analiz sonuçları

Çizelge 4.7 ve Çizelge 4.8’de Kanal-2 ile ilgili değerler görülmektedir. İlk katman sonunda çıkan 1081 tane sahne ikinci katmanda birleştirilerek 14 sahneye kadar bölütlenir ve son katmanda bu değer daha da azalarak 7 sahneye iniyor. Bu sonuçlar gerçek verilerle karşılaştırıldığında 99% başarılı olduğu gözükür. Burada da bazı haberlerin uzun sürmesi birden fazla S-V çiftinin olması başarıyı etkilemiştir. Fakat son katman bu durumları önemli ölçüde azaltmıştır.

#### 4.1.4. Kanal-3 Haber Programı İçin Değerlendirme

Kanal-3 haber programları diğer değerlendirilen kanallara göre daha düzenli bir yapıya sahiptir. Fakat burada da haberlere ayrılan süreler farklılık gösterebiliyor. Günün önemli bir veya iki haberi daha uzun ve detaylı anlatılırken diğer haberlerin üstünden daha kısa sürede geçilir. Bu durumda ilk uzun haber 2. katmanın sonucuna göre tek bir haber olmasına rağmen bir çok habermiş gibi analiz edilmiş olur. Bu durumu yazı bölgesi tanıma katmanı önemli ölçüde tanıyarak aynı habere ait olan sahneleri birleştirir. Ayrıca kanal-3 haber programlarında spiker ilk olarak bütün haberlerin kısa özetinin üzerinden geçer. Bu da sisteme göre S-V çifti olarak algılandığı için bir haber olarak işaretlenmiş olur. Şekil 4.4'de kanal-3'ün örnek çerçeveleri görülmektedir.



Şekil 4.4: Kanal-3 haber örnek çerçeveleri

|                          |                     |
|--------------------------|---------------------|
| Toplam Süre              | 48 dakika 28 saniye |
| Dosya Boyutu             | 337 Mb              |
| Toplam Çerçeve Sayısı    | 72892               |
| Çerçeve Frekansı (FPS)   | 25                  |
| Resim Boyutları (piksel) | 384x240             |
| Video Tipi               | MPEG-4              |

Çizelge 4.9: Kanal-3 örnek 1 video özellikleri

|                                |             |
|--------------------------------|-------------|
| İşlenen Çerçeve Sayısı         | 40000       |
| 1. Katman Haber Sayısı         | 275         |
| 2. Katman Haber Sayısı         | 23          |
| 3. Katman Haber Sayısı         | 13          |
| Gerçek Haber Sayısı            | 9           |
| Doğru Bulunan Haber Geçişleri  | 9           |
| Yanlış Bulunan Haber Geçişleri | 4           |
| Doğru Kabul Oranı              | $9 / 9 = 1$ |

|                           |                    |
|---------------------------|--------------------|
| <b>Yanlış Kabul Oranı</b> | 4 / 40000 = 0.0001 |
| <b>Başarı Oranı (%)</b>   | 99                 |

**Çizelge 4.10: Kanal-3 örneği 1 analiz sonuçları**

Çizelge 4.9 ve Çizelge 4.10’de Kanal-2 ile ilgili değerler görülmektedir. İlk katman sonunda çıkan 275 tane sahne ikinci katmanda birleştirilerek 23 sahneye kadar düşürülür ve son katmanda bu sahneler yazı benzerliğine göre birleştirilerek 13 sahneye inmiş olur. Bu sonuçlar gerçek verilerle karşılaştırıldığında başarı oranı 99% olarak gözükmemektedir. Burada yanlış tanınan geçişler genellikle telefon konuşmaları olduğu için sistemin başarısı olumsuz yönde etkilemiştir.

#### 4.1.5. Kanal-3 Haber Programı İçin Değerlendirme

Kanal-3 haber programları bir önceki bölümde de bahsedildiği gibi daha düzenli bir yapıya sahiptirler. Fakat burada da başarıyı etkileyen farklı durumlar söz konusudur. Örneğin haber programı arasında reklamlar olabilir. Yaklaşık programın ilk 30 dakikasından sonra 2 dakikalık bir reklam arası verilir. İkinci katman analiz yaparken spikerin olduğu ve olmadığı sahneleri araştırdığı için bu durumda reklam videosu da haber videosu olarak algılanmış olur. Diğer bir durum da bazı haberler arası kanal-3 haber logosu gösteriliyor. Bu sahne süresi 15-20 saniye sürdüğü için bölütlemedeki başarıya etkisi düşüktür. Şekil 4.5’de kanal-3’ün örnek çerçeveleri görülmektedir.



**Şekil 4.5: Kanal-3 haber örnek 2 çerçeveleri**

|                                 |                     |
|---------------------------------|---------------------|
| <b>Toplam Süre</b>              | 39 dakika 44 saniye |
| <b>Dosya Boyutu</b>             | 276 Mb              |
| <b>Toplam Çerçeve Sayısı</b>    | 59811               |
| <b>Çerçeve Frekansı (FPS)</b>   | 25                  |
| <b>Resim Boyutları (piksel)</b> | 384x240             |
| <b>Video Tipi</b>               | MPEG-4              |

Çizelge 4.11: Kanal-3 örnek 2 video özellikleri

|                                       |                       |
|---------------------------------------|-----------------------|
| <b>İşlenen Çerçeve Sayısı</b>         | 59800                 |
| <b>1. Katman Haber Sayısı</b>         | 801                   |
| <b>2. Katman Haber Sayısı</b>         | 39                    |
| <b>3. Katman Haber Sayısı</b>         | 18                    |
| <b>Gerçek Haber Sayısı</b>            | 14                    |
| <b>Doğru Bulunan Haber Geçişleri</b>  | 14                    |
| <b>Yanlış Bulunan Haber Geçişleri</b> | 4                     |
| <b>Doğru Kabul Oranı</b>              | $14 / 14 = 1$         |
| <b>Yanlış Kabul Oranı</b>             | $4 / 59800 = 0.00006$ |
| <b>Başarı Oranı (%)</b>               | 99                    |

Çizelge 4.12: Kanal-3 örneği 2 analiz sonuçları

Çizelge 4.11 ve Çizelge 4.12’de Kanal-2 ile ilgili değerler görülmektedir. İlk katman sonunda çıkan 801 tane sahne ikinci katmanda birleştirilerek 39 sahneye kadar azalır ve son katmandaki analiz sonucu bu değer 18 sahneye kadar azalır. Bu sonuçlar gerçek verilerle karşılaştırdığımızda 99% başarı gözükür. Yanlış tanımayı etkileyen durumlar burada da telefon konuşmalarının olmasıdır.

#### 4.2. Test Sonuçlarının Değerlendirilmesi

Elde edilen test sonuçları değerlendirildiğinde sistemin başarılı bir şekilde haber geçişlerini bulduğu anlaşılmaktadır. Bütün örneklerde 99% başarı sağladığı görülmüştür. Bunun sebebi farklı katmanların farklı analiz yöntemleri kullanarak sahneler arası benzerlikleri çıkarması ve bu çıkarımları kullanarak benzer sahneleri birleştirmesidir. Her eklenen katman önceki katmandan elde edilen sonuçları kullanarak bağımsız olarak çalışmaktadır. Genel amaçlı katmanların yanısıra haber programlarının tamamen farklı formatta olması durumunda televizyon kanalına özel olarak tasarlanmış katmanlar da sisteme eklenebilir.

Elde edilen test sonuçları değerlendirildiğinde bölütlemeyi etkileyen bazı faktörler aşağıdaki gibi gruplanmıştır.

#### 4.2.1. Yüz Tanımadaki Hatalar

Sistemin başarısını etkileyen en önemli unsurlardan biri yüz tanımadaki başarısız sonuçlardır. Geliştirilen sistemin ikinci katmanında spiker tanıma işlemi yapılmaktadır. Spiker yüzü bulunamadığında sistemin başarısı düşük çıkacaktır. Dolayısıyla yüz bulma başarısını etkileyen etkenler sistemin de başarısı etkilemiş olacaktır. Tanımayı güçleştiren durumlara aşağıdaki örnekleri verebiliriz:

1. Saç modelinde yüzü kapatabilecek değişiklikler
2. Bazı spikerlerin gözlük kullanması, gözlükte meydana gelen yansımalar
3. Spikerlerin konuşurken fazla baş hareketi yapması, baş açısının önemli derecede değişmesi
4. Ortamdaki ışık değişiklikleri
5. Spikeri çeken kameranın açısının her zaman tam karşıdan çekmemesi, farklı açılardan, özellikle yandan çekim yapılması

Şekil 4.6’da ilkde resim yüzün tam karşıdan görünmemektedir. İkinci resimde ise yine yüz karşıdan görünmemektedir ve saç yüzü kapamaktadır.



Şekil 4.6: Yüz tanımanın başarısız olduğu çerçeve örnekleri

#### 4.2.2. Haber İçeriğinin S-V Yapısına Uymaması

İkinci katmanın temel analiz yapısı sahneler arasından S-V çiftleri bulmaktır. Dolayısıyla ilk olarak spiker sahnesi bulunur ve ondan sonraki sahnenin de haberle ilgili video görüntüsü olduğu varsayılır. Çünkü bir çok haber bu şekilde anlatılır. Fakat bazı haberler çok kısa oldukları için ve görüntüsü olmadığı zaman sadece spiker konuşması ile anlatılır. Spiker haberle ilgili konuştuktan sonra başka bir habere geçerek devam eder. Eğer haber geçişlerinde alt yazılar ile yeni haber ile ilgili bilgi veriliyorsa, 3. katmanda alt yazı farklılıkları değerlendirilerek bu problem çözülmüş olur. Şekil 4.7’de ardışık iki çerçevede haber geçişi görülmektedir.



Şekil 4.7: İki haber arasında haber videosu olmayan örnek çerçeveler

#### 4.2.3. Haber Özetleri

Bazı haber programlarında spiker haberlere geçmeden önce anlatılacak haberler kısaca özetlenir. Bu durum da aslında bir S-V çifti şeklinde bölütlenir. Dolayısıyla haber özetleri görüntüsünde spiker yer almıyorsa bu sahneyi bir haber sahnesi olarak bölütlüyor.



Şekil 4.8: Haber özetlerine örnek ardışık çerçeveler

#### 4.2.4. Güncel Haberlerin Aynı Haber Programında Tekrar İşlenmesi

Güncel haberlerle ilgili son gelişmeleri vermek için aynı haber programı içinde canlı bağlantılar ile farklı zamanlarda aynı konudan bahsedilebilir. Örneğin uçak kazası gibi önemli bir haber olduğunda, olayla ilgili yeni bir haber geldikçe haber programı kesilip bu haberlerin verilmesi gerekebilir. Bu da aynı haberin haber programının farklı bölümlerinde yer alabileceği anlamına gelir. Böyle bir durum olduğu zaman bu ikinci katman için farklı bir haber demektir. Dolayısıyla da daha önce anlatılan haberle birleştirilmesi mümkün değildir. Bu durumu spikerin konuşmasının ve alt yazıların içeriğini anlayarak değerlendirecek bir katman tasarlanmalıdır.

## 5. SONUÇ

Bu çalışmada bir haber programı videosundaki haber geçişlerini bulan bir sistem tasarlanmış ve gerçekleştirilmiştir.

Televizyon kanallarında hergün yüzlerce haber yayınlanmaktadır. Bu haberler, tarihi değeri olabilecek bilgiler içermektedir. Gelişen yazılım ve donanım imkanları ile çok büyük verilerin arşivlenmesi mümkün hale gelmiştir. Ancak arşivlenen veriye hızlı bir şekilde ulaşabilmek veriyi saklamaktan daha önemli ve daha çok çaba gerektiren bir konudur. Bir konu ile ilgili bir habere ulaşmak için arşivdeki mevcut bütün verinin saatlerce taranması efektif bir çözüm değildir. Haberlere içeriğine göre erişimi sağlama konusunda ciddi çalışmalar yapılmasına ihtiyaç vardır. Bu çalışmada bu işlemin bir ön adımı olmak üzere haber geçişlerini bulan bir sistem tasarlanmış ve gerçekleştirilmiştir.

Haber geçişlerinin nasıl yapılması gerektiğine karar vermek için Türk kanallarındaki mevcut haber programları incelenerek karakteristik özellikler belirlenmeye çalışılmıştır. Bu değerlendirme sonucunda haber programlarında genel olarak önce spikerin haberi tanıttığı, daha sonra haber ile ilgili video görüntülerin görüldüğü ve sonra bir sonraki habere geçildiği görülmüştür. İstisnai durumlar önceki bölümlerde incelenmiştir.

Haber geçişlerini bulmak için gerçekleştirilen bölütleme işleminin katmanlar şeklinde yapılması öngörülmüştür. Bunun için her katmanda farklı analizlerin yapıldığı üç katmanlı bir sistem geliştirilmiştir. İlk katmanda renk ve kenar değişimi benzerlikleri değerlendirilerek aynı habere ait olduğu düşünülen çerçeveler birarada olacak şekilde gruplama işlemi yapılmıştır. İkinci katmanda, her grup içinde spiker yüzü tanıma işlemi gerçekleştirilmiştir. Gruptaki çerçevelerden belli bir miktarında spiker yüzü bulunduğunda diğer çerçevelere bakılmayarak işlem zamanı kısaltılmıştır. Son katmanda çerçevelerin yazı bölgeleri değerlendirilerek aynı habere ait olup olmadıkları değerlendirilmiştir.

Geliştirilen sistemin haber geçişlerini bulma başarısı farklı haber kanallarından alınan örnek haber programları üzerinde denenerek değerlendirmiştir. Sistemin haber geçişlerini bulma başarısının ortalama %99 olduğu görülmüştür.

Sistemin katmanlı bir yapıda tasarlanmasından ötürü, farklı katmanlar da eklenerek bölütleme sisteminin başarısının artırılması mümkündür. İlerki çalışmalarda, haber geçişlerine göre gruplanmış haberlerin içerikleri değerlendirilerek bir haber programını tamamen otomatik olarak bölütleyen, içeriğine göre arşivleyen ve bu sayede kısa sürede arama yapılabilinen sistemler tasarlanması mümkün olacaktır.

**KAYNAKLAR**

- [1] Rafael C. Gonzales, Richard E. Woods (2002), Digital Image Processing Second Edition, Prentice Hall
- [2] Sameer Raj Maskey, Julia Hirschberg, Automatic Summarization of Broadcast News using Structural Features, Columbia University, New York
- [3] N. O'Connor, C. Czirjek, S. Deasy, S. Marlow, N. Murphy, A. Smeaton, News Story Segmentation in the Fischlar Video Indexing System, Dublin City University
- [4] Wei-Hao Lin, Alexander Hauptmann, News Video Classification Using SVM-based Multimodal Classifiers and Combination Strategies, Carnegie Mellon University
- [5] Jae-Gon Kim, Hyun Sung Chang, Kyeongok Kang, Munchurl Kim, Jinwoong Kim, Hyung-Myung Kim, Summarization of News Video and Its Description for Content-based Access, Electronics and Telecommunications Research Institute, Information and Communications University, Korea Advanced Institute of Science and Technology, Daejeon, Korea
- [6] M. Bertini, A. Del. Bimbo, P. Pala, Content-based indexing and retrieval of TV news, Dipartimento di Sistemi e Informatica, Universita di Firenze, Via Santa Marta 3, I-50139, Firenze, Italy
- [7] Linda G. Shapiro, George C. Stockman, Computer Vision, Prentice Hall
- [8] Jolliffe I.T.(2002) Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, XXIX, 487 p. 28 illus.
- [9] Jon Shlens, A Tutorial on Principal Component Analysis.
- [10] Rainer Lienhart, Comparison of Automatic Shot Boundry Detection Algorithms, Microcomputer Research Labs, Intel Corporation, Santa Clara,
- [11] Canny, J., (1986)A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714
- [12] Sobel, I., Feldman,G., (1968), "A 3x3 Isotropic Gradient Operator for Image Processing", presented at a talk at the Stanford Artificial Project.
- [13] Haar A. Zur (1910), Theorie der orthogonalen Funktionensysteme, Mathematische Annalen, 69, pp. 331-371.
- [14] N. Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 9: 62–66.
- [15] Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence 2 (12): 1137–1143. <http://citeseer.ist.psu.edu/kohavi95study.html>. (Morgan Kaufmann, San Mateo)

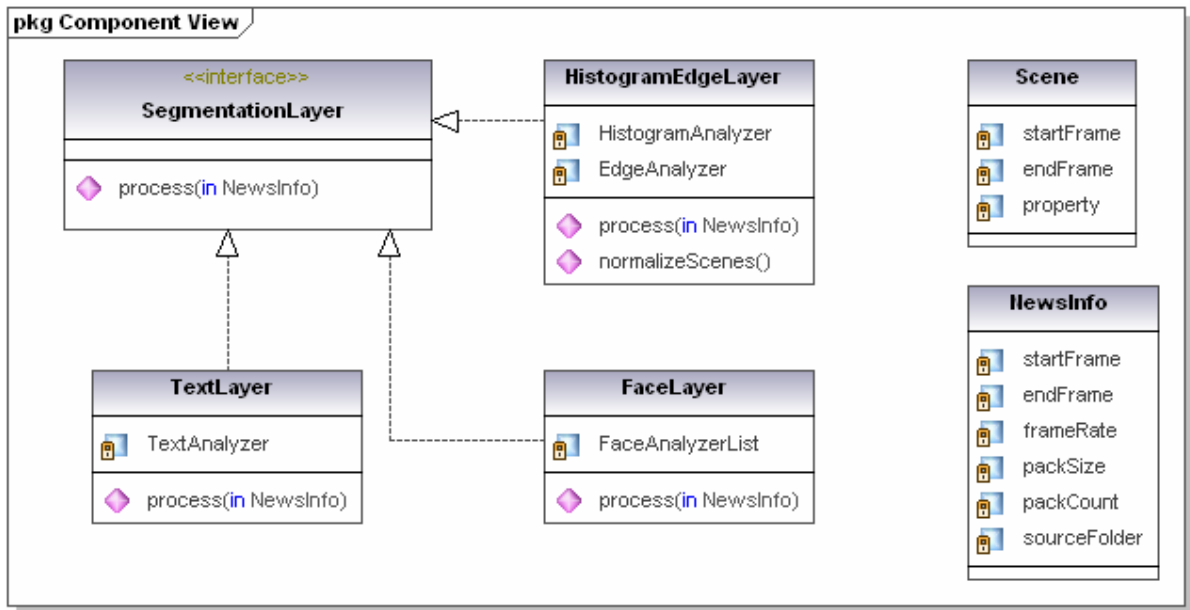
- [16] Wendy S. Yamhor July (2000), Analysis of PCA and Fisher Discriminant-Based Image Recognition Algorithms –Technical Report CS-00-103, Colorado State University
- [17] Thomas Heseltine, Evaluation of Image Pre-processing Techniques for Eigenface Based Face Recognition, University of York
- [18] M. Turk, A. Pentland (1991), "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, 3(1), pp. 71-86.
- [19] D. Swets, J. Weng(1996), "Using Discriminant Eigenfeatures for Image Retrieval", IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8), pp. 831-836.
- [20] A. Martinez, A. Kak (2001), "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228-233.

## EKLER

### Ek 1 UML Diagramları

Sistemin tamamı Java dilinde kodlanmıştır. Görüntü işleme yöntemleri için C veya C++ dillerinde geliştirilmiş kütüphaneler bulunmaktadır. Fakat Java dilinde görüntü işleme üzerine yaygın kullanılan kütüphaneler olmadığı için gerekli bütün yöntemler için kod yazılarak bir kütüphane haline getirilmiştir. Yalnız görüntüde yüz bulmak için OpenCV kütüphanesinin Java’da kullanılmasını sağlayan özel bir kütüphane kullanılmıştır.

Bu bölümde geliştirilen java objeleri üzerinden sistem tasarımının nasıl geliştirildiği incelenecektir. Sistem tasarımı bölümünde katmanlardan bahsedilmişti ve üç farklı katmanların tasarımı anlatılmıştı. Aşağıdaki Katmanlar UML Diagramı, katman java sınıflarını ve onların kullandığı yardımcı sınıfları göstermektedir.



Katmanlar UML Diagramı

#### NewsInfo.java

Bu java sınıfı katmanların analiz aşamasına başlamadan önce ihtiyaç duyduğu bilgileri içerir. Geliştirme aşamasında her defasında bütün videoyu baştan sona analiz etmek uzun süren bir işlem olduğu için burada **startFrame** ve **endFrame** adlı iki özellik sayesinde katmanlara hangi bölgeyi analiz etmesi gerektiği söylenebilir. Bütün resimleri tek dizin altında toplamamak için onluk gruplar halinde dizinlere dağıtılır. Her dizinde kaç tane resim olacağı

**packSize** değeri tutulur ve kaç tane dizin olduğu da **packCount** değeri saklanır. Analiz aşamasında her bir çerçeve için bu değerler kullanılarak gerçek resim dosyasının yeri tespit edilir. **sourceFolder** fragman resimlerinin hangi lokasyonda olduğunu belirtir. **frameRate** ise haber videosunun çerçeve hızını göstermektedir.

### Scene.java

Scene sınıfı ise tek bir sahne ile ilgili bilgileri tutmaktadır. Bu bilgiler sahnenin başlangıç ve bitiş fragmanları ve sahneye ait özelliklerdir. Her katman giriş olarak bu sınıftan oluşan bir diziyi alır ve yine sonuç olarak bu sınıftan oluşan bir dizi oluşturur. En son katmanın oluşturduğu sahne dizi ise artık bölütlenmiş haberleri içermektedir.

### SegmentationLayer.java

Bu nesne bir sınıf değil, bütün katmanlar için tanımlayıcı bir kalıptır. Java’da böyle nesnelere **interface** denilir. Bu nesneler yazılacak sınıfların hangi fonksiyonelleri içermesi gerektiğini tanımlar. Yapılmış olan tasarımda tek fonksiyonelite vardır o da **process**’tir. Yani giriş sahneleri analiz ederek sonuç sahneleri oluşturmaktır. Bu işlem giriş verisi olarak NewsInfo nesnesi alır. Bütün eklenen katmanlar buradaki kalıba uymak zorundadırlar.

### HistogramEdgeLayer.java

Bu java sınıfı tasarımda bahsedilen ilk katmanı ifade etmektedir. Sistemin yapısında genel olarak katmanlar ve analiz eden sınıflar var. Bir katman birden fazla analiz nesnesi içerebilir. Aynı zamanda farklı katmanlar da aynı analiz nesnelerini ortaklaşa kullanabilir. Sistem bunu destekliyor çünkü katmanlar ve analiz nesneleri bir birinden bağımsız geliştirildi. İstenildiği zaman katmanlara yeni analiz sınıfları eklenebilir. Böyle bir tasarım ilerde yeni analiz sınıflarının katmanlara kolayca eklenmesini sağlayacaktır. Aynı zamanda analiz nesnelerinin başarısı ölçebilmek için bazılarını kolayca devre dışı bırakmamızı da sağlıyor.

**HistogramEdgeLayer** sınıfında iki tane analiz nesnesi kullanılmıştır:

- HistogramAnalyzer
- EdgeAnalyzer

Bunların her ikisi de sahne geçişlerini yakalamakta kullanılır. İlk olarak histogram analizi çalışır daha sonra onun yakalayamadığı durumlarda kenar piksel analizi yapılarak sahne geçişleri yakalanmış olur. Bu geçişler yakalandıktan sonra giriş fragmanları sahnelere ayrılarak bir sonraki katman için işlenmeye hazır hale getirilir.

### FaceLayer.java

Bu katman spiker tanıma katmanıdır. Dolayısıyla burada yüz tanıma işlemleri gerçekleştirilir.

Bu katman tek analiz nesnesi vardır:

- FaceAnalyzer

Fakat haber videolarında her zaman aynı spiker konuşmadığı için birden fazla spiker tanınmanın sağlanması lazım. Bu yüzden bu analiz nesnesinden bir den fazla kullanılır. Her spiker için ayrı ayrı **FaceAnalyzer** nesneleri liste halinde katmanda yer almaktadır. Dolayısıyla analiz aşamasında bulunan yüzler bütün yüz tanıma analiz nesnelerinde işlenir ve herhangi birisinde tanınırsa bu çerçeve spiker sahnesi olduğu anlaşılır. Bu katmanda ilk katmandan gelen sahneler S ve V şeklinde işaretlenir. S ile işaretlenenler spikerin olduğu sahneler ve V'ler ise spiker olmayan çerçevelerdir. Her S-V çifti bir haber olarak kabul edebilir. Bu yüzden her S-V çiftleri son olarak birleştirilir ve tek bir sahne haline getirilir.

### TextLayer.java

Bu katman çerçeveler arasındaki yazı özelliklerini inceler. Bu katman son katman olduğu için önceki katmanlarda oluşan sonuçları kendi analiz yöntemleri ile birleştirerek sonuç bölütlemeyi oluşturur.

Katmanlarla ilgili geliştirmeleri inceledikten sonra onların kullandığı ve bağımsız olarak istenilen yerde kullanılabilen analiz nesnelerinden bahsedelim.

### Analyzer.java

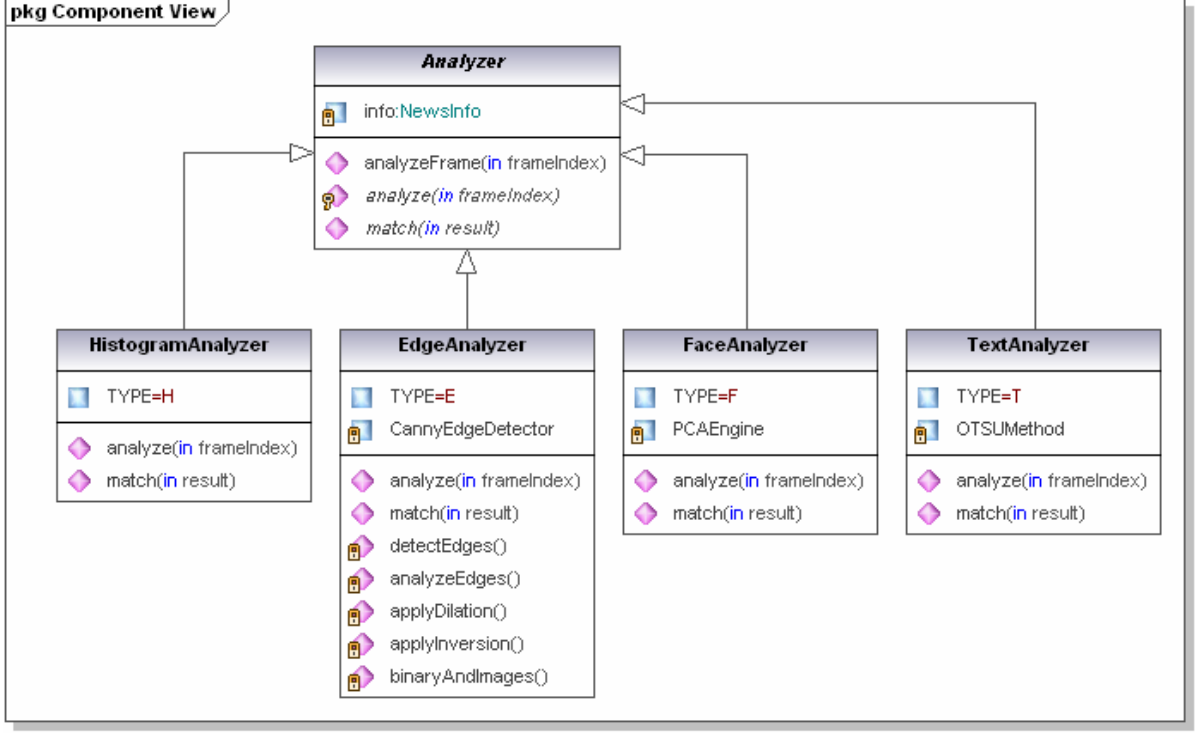
Bu sınıf bir analiz nesnesinin nasıl olması gerektiğini anlatan genel bir Java nesnesidir. Bu nesneler soyut olduğu için onlara **Abstract Class**'lar deniliyor. Çünkü bu nesnelerden fiziksel objeler oluşturulamaz. Bu sınıflar sadece kullanılarak fiziksel iş yapan alt nesneler geliştirilir.

Burada geliştirilmesi istenen iki ana fonksiyonelite var:

- *analyze*
- *match*

**Analyze** işlemi verilmiş çerçeve için gereken analizlerin yapıldığı bölümdür. Fiziksel sınıflar bu bölümde yapılması gereken analiz kodlarını içerecektir. **match** işlemi ise analiz bittikten

sonra çıkan sonuçları değerlendirmek için kalıp bölgesidir. Tabiki her analiz sınıfı bu kısmı kendine özgü bir şekilde dolduruyor olacaktır. **match** fonksiyonun boolean(doğru veya yanlış) cevabı oluşmaktadır yani evet veya hayır türünden sonuç üretmektedir. Örneğin yüz tanıma yaparken çerçevedeki yüzün spiker olup olmaması cevabını analiz nesnesi bize evet veya hayır şeklinde ifade etmektedir.



Analiz UML Diagramları

### HistogramAnalyzer.java

**HistogramAnalyzer** sınıfı fiziksel sınıf olup renk değişimi ilgili analizlerin geliştirildiği sınıftır. Bu sınıf verilmiş çerçeve için histogram bilgisi çıkararak bu bilginin bir sonraki çerçevenin histogramı ile karşılaştırır ve mesafe sonucunu sayı olarak geri döndürür. Daha sonra **match** fonksiyonelitesi kullanılarak bu değerin sahne geçişini belirtip belirtmediğini sorgulanır. Histogram farkı RGB olarak her üç ekseninde ayrı-ayrı hesaplanır.

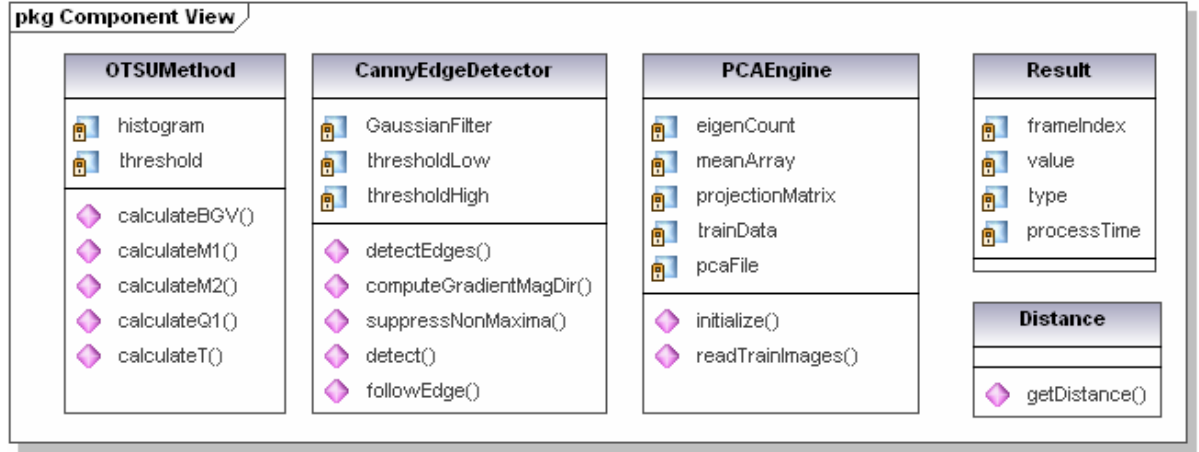
### EdgeAnalyzer.java

Bu sınıf kenar piksellere göre sahne geçişini bulan analiz nesnesidir. Analiz işleminde verilmiş çerçeve ile bir sonraki çerçeve arasında fark değeri hesaplanır ve bu değer **match** işlemi sonucunda sahne geçişi olup olmadığına karar verilmesinde kullanılır. Kenar pikselleri belirlemek için başka bir sınıf, **CannyEdgeDetector** kullanılmıştır.

### FaceAnalyzer.java

Yüz tanıma veya spiker tanıma işlemleri bu sınıfta yapılmaktadır. Bu sınıf öncekilerden farklı olarak bir sonraki çerçeve ile ilgili bir işlem yapmamaktadır. Sadece verilmiş çerçeve üzerinde spiker yüz bulmayı hedefler. Daha önce de belirtildiği gibi katmanda birden fazla **FaceAnalyzer** nesneleri olacaktır. Çünkü her spiker için eğitim verisi farklı ve tanıma için belirlenmiş eşik seviyesi de farklıdır. Dolayısıyla her **FaceAnalyzer** nesnesi kendi içinde eğitim verisinin olduğu yeri ve eşik seviyesini tutmaktadır. Tasarıma baktığımızda da her sistemde tanımlı spiker sayısı kadar analiz nesnesinin katmana eklenmiş olması gerekmektedir.

Son olarak da katman ve analiz nesnelerinin kullandığı yardımcı nesnelerden bahsedelim.



Yardımcı bileşenler UML Diagramı

### TextAnalyzer.java

Bu analiz sınıfı çerçevelerin yazı özelliklerini karşılaştırır. Bunun için OTSU sınıfını kullanarak resimi siyah-beyaz rene dönüştürür daha sonra ise mesafe hesabı yaparak her iki çerçevenin aynı habere ait olup olmadığına bakar.

### Result.java

Her analiz sınıfı işlemini bitirdikten sonra **Result** sınıfından bir nesne üretir. Burada yaptığı analizin sonuçlarını tutar. **frameIndex** sonucun hangi fragmana ait olduğunu gösterir. **value** değeri analiz sonucunda çıkan mesafe değeridir. **type** değeri her hangi analiz sınıfının sonucu olduğunu gösteren bir işarettir. **processTime** ise analiz ne kadar zamanda tamamlandığını gösterir.

**Distance.java**

Çeşitli mesafe hesaplarının olduğu sınıftır. Bütün mesafe hesaplarında Öklit mesafe yöntemi kullanılmıştır.

**CannyEdgeDetector.java**

Bu sınıf EdgeAnalyzer sınıfı tarafından fragmandaki kenar pikselleri bulmak için kullanılır. Kendi içerisinde Gauss filtersi ile ilk önce fragman resmi üzerinde yumuşama gerçekleştirir. Daha sonra Canny yöntemini uygulayarak kenar pikselleri bulur.

**PCAEngine.java**

Bu sınıf bütün yüz tanıma işlemlerini yürütmektedir. Sistemin eğitilmesinden eigen vektörlerin hesaplanmasına kadar bütün işlemler burada geliştirilmiştir. PCA işlemleri adımlar halinde yapılır ve eigen vektörlerinin bulunması için hazır **Jama** kütüphanesi kullanılmıştır.

**OTSUMethod.java**

Bu sınıf bütün yüz tanıma işlemlerini yürütmektedir. Sistemin eğitilmesinden eigen vektörlerin hesaplanmasına kadar bütün işlemler burada geliştirilmiştir. PCA işlemleri adımlar halinde yapılır ve eigen vektörlerinin bulunması için hazır **Jama** kütüphanesi kullanılmıştır.

**Kullanılan Kütüphaneler**

Java'da video analiz için kaynak eksikliği yüzünden bir çok temel analiz yöntemleri sıfırdan geliştirildi. Bunların dışında hazır olarak kullanılan kütüphaneler aşağıdaki bölümlerde açıklanmıştır.

**Java Media Framework 1.2**

JMF Sun tarafından geliştirilen video ve ses dosyalarını okumak ve yazmak için kullanılan nesneleri içeren bir kütüphanedir. Özellikle video dosyalarını okumak için içinde bir çok codec bilgisi içerir. Bu çalışma kapsamında video dosyalarını okumak ve yazma için bu kütüphane kullanılmıştır.

**Jama 1.0.2**

Jama açık kod olarak yayınlanan matris işlemleri için kullanılan bir kütüphanedir. Matris çarpımlarının yanı sıra PCA için kullanılan eigen vektörlerin hesaplanması için de kullanılmıştır.

**OpenCV**

OpenCV normalde C++ dilinde yazılan görüntü işleme için kullanılan çok yaygın bir kütüphanedir. Fakat kodlar C++ dilinde yazıldığı için başka dillerden kullanılması mümkün olmuyor. Bu çalışma kapsamında Java'nın sunduğu JNI (Java Native Interface) kullanılarak FAINT kütüphanesi yardımıyla Java'dan C++ kodları çağrılabilmiştir. OpenCV kütüphanesi kullanılarak resimdeki yüzleri bulmak için kullanılmıştır.

## ÖZGEÇMİŞ

|                   |                      |                              |
|-------------------|----------------------|------------------------------|
| Doğum tarihi      | 09.09.1984           |                              |
| Doğum yeri        | Zaqatala, Azərbaycan |                              |
| İlk okul          | 1991-1996            | S.Cabbarov adına 252 No Okul |
| Orta okul ve Lise | 1996-2001            | Bakü Özel Türk Lisesi        |
| Lisans            | 2001-2005            | Marmara Üniversitesi         |

## Başarılar

|                                                 |      |                     |
|-------------------------------------------------|------|---------------------|
| Azərbaycan Bilgisayar Olimpiatları mansiyon     | 2000 | Bakü, Azərbaycan    |
| Azərbaycan Bilgisayar Olimpiatları altın medalı | 2001 | Bakü, Azərbaycan    |
| Uluslararası Bilgisayar Olimpiatları iştirakçı  | 2001 | Tampere, Finlandiya |