



T.C.

ALTINBAŞ UNİVERSİTESİ
Bilişim Teknolojileri

**GÖZ HAREKETLERİNE DAYALI
BEYİN BİLGİSAYARI ARAYÜZÜ
TASARIMI**

Engin KOÇ

Master Tezi

Danışman
Prof. Dr. Oğuz Bayat

Istanbul,2019

GÖZ HAREKETLERİNE DAYALI BEYİN BİLGİSAYARI ARAYÜZÜ TASARIMI

Engin Koç

Bilişim Teknolojileri Bölümü

Fen Bilimleri Enstitüsü'ne

Lisansüstü Derece

tezi olarak sunulmuştur.

ALTINBAŞ ÜNİVERSİTESİ

2019

Bu çalışma tarafımızca incelenmiş olup, kapsam ve kalite açısından Yüksek Lisans tezi olmaya yeterli bulunmuştur.

Dr. Öğr. Üyesi

Adil Deniz DURU

Eş Danışman

Prof. Dr. Oğuz BAYAT

Danışman

İnceleme Komitesi Üyeleri (İlk isim jüri başkanına, ikinci isim tez danışmanına aittir.)

Prof.Dr. Oğuz BAYAT

Fen Bilimleri Enstitüsü

Altınbaş Üniversitesi

Asst.Prof.Dr. Doğu Çağdaş ATILLA

Fen Bilimleri Enstitüsü

Altınbaş Üniversitesi

Dr.Öğr. Üyesi Oğuz ATA

Mühendislik ve Doğa

Bilimleri Fakültesi

Altınbaş Üniversitesi

Dr.Öğr. Üyesi Adil Deniz DURU

Spor Bilimleri Fakültesi

Marmara Üniversitesi

Dr.Öğr. Üyesi Dilek Göksel DURU

Mühendislik ve Mimarlık

Fakültesi

Arel Üniversitesi

Bu çalışma bir Yüksek Lisans tezinin tüm gerekli şartlarını taşımaktadır.

Doç. Dr. Oğuz ATA

Bölüm Başkanı

Prof. Dr. Oğuz BAYAT

Enstitü Müdürü

Fen Bilimleri Enstitüsü onayı: ___/___/___

Bu dökümandaki tüm bilgilerin akademik kural ve etiğe bağılı kalınarak yazıldığını ve tez yazım kuralları kapsamında bu çalışmada bulunan ve özgün olmayan bütün bilgi ve materyallerin referanslandırıldığını temin ederim.

Engin KOÇ



ÖZET

GÖZ HAREKETLERİNE DAYALI BEYİN BİLGİSAYARI ARAYÜZÜ TASARIMI

Engin Koç,

Yüksek Lisans, Bilişim Teknolojileri, Altınbaş Üniversitesi,

Danışman: Prof. Dr. Oğuz BAYAT

Eş Danışman: Dr. Öğr. Üyesi Adil Deniz DURU

Tarih: Mayıs, 2019

Sayfa Sayısı: 141 Sayfa

Modern teknoloji ile birlikte insanların göz hareketlerini inceleyerek uyarıcılara karşı vermiş oldukları tepkiler takip edilebilir hale gelmiştir. Bu takip yöntemlerinden biri de Göz İzleme (Eye-Tracking) tekniğidir. Bu teknik sayesinde sağlık, savaş sanayi, sivil havacılık, web tasarımı, dijital medya vb bir çok alanda hayatı daha kolay hale getirilebilecek sistemler hakkında çalışmalar yapılmaktadır. Bu çalışma amacı, göz izleme teknolojisinin temel yönleri ve konuları, gelişim süreçlerinin incelenmesi, göz izleme uygulaması geliştirmek için pratik rehberlik , göz izleme kullanarak kurgulanması ve insan makine arayüzü tasarlanmasıdır. Bu amaçla C# tabanlı göz hareketlerini izlemeye olanak sağlayan bir yazılım geliştirilmiştir. Eye-tribe cihazı ile C#'da birinci çalışmamız cihaza bağlanma ve göz hareketlerini izleyerek veri akışı sağlamak olmuştur. Burada elde edilen veriler text türünde bir belgeye yazılmıştır. Çalışmamızda ikinci aşama olarak deneysel ortamda ekranda kullanıcıya sorulan sorulara eye-tribe cihazı ile cevap verebilme kabiliyeti analiz edilmiştir. Ortaya çıkan veriler database yazılmıştır. Sorular ve cevaplar arasında geçen süreler tablo olarak çalışmamızda belirtilmiştir. Geliştirilen çalışmada alınan verilere istinaden üçüncü aşama olarak random kullanıcılara koordinatları sürekli olarak değişen butona eye-tribe cihazı ile göz izleme tekniği ile C# uygulama yazılmıştır. Geliştirilen bu uygulamada kullanıcılar gözleri ile ekranda farklı konumlarda çıkan butona göz izleme yöntemi ile bakarak cevap verilmesi istenmiştir. Elde edilen veriler database kayıt edilmiştir. Kayıt edilen bu verilerde kullanıcıların göz izleme yöntemi ile verdikleri tepkiler analiz edilmiştir. Ayrıca çalışmaların ilerlemesi sonucunda gelinen noktadan pazarlama alanında bu tekniğin kullanımına dair bir çalışma daha gerçekleştirilmiştir. Kullanıcının göz izleme çalışmamızda ekranda baktıkları noktaların

sıcaklık haritası çıkarılmıştır. Burada koordinat sistemi kullanılarak ekranda hangi alanlara daha çok odaklandığı tespit edilmiştir. Tespit edilen bu noklara daire şeklinde ekranda işaretlenmiştir. Son çalışmamızda ise hareket yetisini katbetmiş insanlara göz izleme takibi yapılarak ekranda sanal klavyede harflerden cümleler kurması kurgulanmıştır. Ekranda yer alan her bir harf koordinat içermektedir. Bu koordinatlar göz izleme cihazı ile aynı noktaya odaklandığı zaman geliştirmiş olduğumuz C# projemiz uyarı sesi vermektedir. Seçilen cümleleri ekrana bakarak kullanıcılardan yazılması istenmiştir. Bu çalışmada veriler yine database yazılmıştır. Database her bir cümlenin başlangıç ve bitiş zamanları analiz edilmiştir.

Anahtar kelimeler: Göz Hareketlerine Dayalı Beyin Bilgisayar Arayüz Tasarımı, İnsan-Bilgisayar Etkileşimi, Kullanışlılık Testi, Göz İzleme, Sanal Klavye



ABSTRACT

DESIGN OF BRAIN COMPUTER INTERFACE BASED ON EYE MOVEMENTS

Engin Koç,

M.S., Information Technologies, Altınbaş University,

Supervisor: Prof. Dr. Oğuz BAYAT

Co Supervisor: Dr. Öğr. Üyesi. Adil Deniz DURU

Date: May, 2019

Pages: 141 Page

With the help of modern technology, people's reactions to stimulants by examining the eye movements have become traceable. One of the follow-up methods is Eye-Tracking technique. In this technique, health, war industry, civil aviation, web design, digital media, etc. are being studied on systems that can be made easier in many areas. The purpose of this study is to examine the basic aspects and subjects of eye tracking technology, to examine the development processes, to develop practical guidance to develop eye tracking practice, to design using eye tracking and to design a human machine interface. For this purpose, software has been developed to enable monitoring of C # based eye movements. Our first work in C # with the eye-tribe device was to provide data flow by connecting to the device and monitoring eye movements. The data obtained here is written in a text document. In the second stage of our study, the ability of the user to respond to the questions asked in the experimental environment with the eye-tribe device was analyzed. The resulting data is written in the database. The times between the questions and the answers are indicated in our study as a table. According to the data obtained in the study, C # application was created by eye tracking technique with eye-tribe device which changed the coordinates continuously to random users as third stage. In this developed application, users were asked to respond by looking at the button with different eye positions. The data obtained were recorded in the database. In this data, the responses of the users with eye tracking method were analyzed. In addition, a further study on the use of this technique in the field of marketing has been carried out as a result of the progress of the studies. In the user's eye tracking study, the temperature map of the points they were looking at the screen was removed. It is determined that the coordinate system focuses more on the areas on

the screen. These points are marked on the screen in circle form. In our last study, the people who have contributed to the movement ability to follow the eye tracking on the virtual keyboard to create sentences from the letters are fictionalized. Each letter on the screen contains coordinates. When these coordinates are focused on the same point as the eye tracking device, our C # project that we developed has been alerted. The selected sentences were asked to be written from the users by looking at the screen. In this study, data is written in the database. The beginning and end times of each sentence were analyzed.

Keywords: Brain Computer Interface Design, Human-Computer Interaction, Usability Testing, Eye Tracking, Virtual Keyboard



İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	V
ABSTRACT.....	vii
TABLO LİSTESİ.....	xı
ŞEKİL LİSTESİ.....	xii
1. GİRİŞ	1
2. GÖZ HAREKETLERİ TAKİBİ VE BİLİŞSEL SÜREÇLER.....	3
2.1. GÖZÜN YAPISI.....	3
2.2. GÖZ HAREKETİ TAKİBİNDE KULLANILAN YÖNTEMLER.....	4
2.2.1. Elektro Okülografi.....	7
2.2.2. Kontak Lens	9
2.2.3. Kızılötesi Okülografi.....	10
2.2.4. Video Okülografi.....	11
2.2.5. Göz Hareketi Takibinde Kullanılan Tekniklerin Karşılaştırılması	16
2.3. GÖZ HAREKETLERİ VE BİLİŞSEL SÜREÇLER	18
2.3.1. Dikkat	18
2.3.2. Algı.....	19
2.3.3. Bağlam Etkisi	20
2.3.4. Örüntü Tanıma	20
2.3.5. Zihinsel İmgeleme.....	20
2.4. GÖZ HAREKETİ VERİLERİ	21
2.4.1. Odaklanma (Fixation)	21
2.4.2. Sıçrama (Saccade).....	22
2.4.3. İzlenen Yol (Scanpath).....	23
2.4.4. Akıcı İzleme (Smooth Pursuit).....	24
2.4.5. Görsel Veriler	24
3. GÖZ İZLEME TEKNOLOJİSİ	25
3.1. WEB UYGULAMALARININ KULLANILABİLİRLİK TESTLERİ	28

3.2. WEB UYGULAMALARININ KULLANILABİLİRLİĞİNDE GÖZ İZLEME TEKNOLOJİSİNİN YERİ	30
3.3. KULLANILABİLİRLİK TESTLERİ İÇİN GÖZ İZLEME METRİKLERİ.....	31
3.4. GÖZ İZLEME TEKNOLOJİSİ VE KULLANIM ALANLARI	34
3.4.1. İnsan Bilgisayar Etkileşimi	34
3.4.2. Pazarlama ve Reklâm	34
3.4.3. Ergonomi ve İşyeri Tasarımı (Human Factors).....	36
3.4.4. Özürlülere Yardımcı Olan Uygulamalar	36
3.4.5. Psikoloji, Psikofizik Ve Nöroloji	37
3.4.6. Grafik, Sanal Gerçeklik Ve Simülatör Uygulamaları	37
4. BEYİN BİLGİSAYARI ARAYÜZÜ	39
5. DENEYSEL ÇALIŞMAYA GİRİŞ	39
5.1. CİHAZ SABİTLENMESİ.....	42
5.2. EYE TRİBE İZLEYİCİ İLK VERİ AKIŞI VE EKRANDA GÖSTERİLMESİ	44
5.3 EYE TRİBE İZLEYİCİ İLE SICAKLIK HARİTASI ÇIKARTMAK.....	46
5.4 EYE TRİBE İZLEYİCİ İLE SORU CEVAP UYGULAMASI.....	49
5.4.1. Database ve Tablo Yapımız	56
5.5 EYE TRİBE İZLEYİCİ İLE OTOMATİK SORU CEVAP UYGULAMASI	57
5.6 EYE TRİBE İZLEYİCİ İLE SANAL KLAVYE UYGULAMASI.....	62
6. SONUÇ VE TARTIŞMA	66
KAYNAKÇA	70
EK A. EYE TRİBE SDK (SOFTWARE DEVELOPMENT KIT) YÜKLENMESİ	78
EK B. 5.2. UYGULAMA KOD ÇIKTISI	79
EK C. 5.3 UYGULAMA KOD ÇIKTISI	83
EK D. 5.4 UYGULAMA KOD ÇIKTISI	87
EK E. 5.5 UYGULAMA KOD ÇIKTISI	104
EK F. 5.6 UYGULAMA KOD ÇIKTISI.....	111

TABLO LİSTESİ

	<u>Sayfa</u>
Tablo 2.1 : Göz Hareketi İzleme Teknolojilerinin Karşılaştırılması	17
Tablo 5.1: Örnek Uygulama Sonuçları	51
Tablo 5.2: Örnek Uygulama Sonuçları	53
Tablo 5.3: Örnek Uygulama Zamanı Sonuçları	55
Tablo 5.4: Örnek Uygulama Sonuçları	59
Tablo 5.5: Örnek Kelime Oluşturma Uygulaması	64
Tablo 6.1: Örnek Cümle ve Analiz Sonuçları.....	67
Tablo 6.2: Örnek Cümle ve Analiz Sonuçları.....	68

ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 2.1 : Gözün Şematik Yapısı	3
Şekil 2.2 : Gözün Karşıdan Görünüşü	4
Şekil 2.3 : Göz Takip Uygulamaları Hiyerarşisi.....	5
Şekil 2.4 : Polarizasyona Göre Bağlantı Şeması a) Göz İçi Polarizasyonu b) Göz Çevresine Elektrotların Yerleştirilmesi.....	8
Şekil 2.5 : Elektrotların Kullanıcının Göz Çevresine Yerleştirilmesi.....	8
Şekil 2.6 : Manyetik Arama Bobin Kontak Lens.....	9
Şekil 2.7 : Sol Göz Üzerine Yerleştirilmiş Kontak Lens	9
Şekil 2.8: Aynalı Kontak Lensler.....	10
Şekil 2.9 : Kızılötesi Okülografi Metodu Çalışma Şekli	10
Şekil 2.10 : Hasta ve Göz Kelime İşlemci Düzeneği.....	11
Şekil 2.11 : Video Ökülografi Tabanlı Teknikler (a) Uzak Göz Takip Cihazı (b) Kafaya Monte Göz Takip Cihazı Konumu	12
Şekil 2.12 : Göz bebeği ve Limbus İzlemek İçin Geliştirilen kafaya Monte Gözlük Tasarımı	12
Şekil 2.13 : Işığın Gözden Yansıması Sonucu Oluşan Birinci, İkinci, Üçüncü ve Dördüncü Purkinje Resimler.....	13
Şekil 2.14 : Yansıyan Işığın Etkisi (a) Işık Kaynağının Göz Üzerinde Oluşturduğu Parıltı (b) Kornea Yansıması.....	14
Şekil 2.15 : Parıltıyerleri (a) Gözişik kaynağına doğru bakarken (b) Göz ışık kaynağının alt kısmına bakarken (c) Gözişik kaynağının sol alt kısmına bakarken.	14
Şekil 2.16 : Göz Bebeği ve Kornea Yansıması Sistemi.....	15
Şekil 3.1: İlgi Alanlarına Ayrılmış Örnek Bir Web Sayfası (www.zaccheoatrental.com) ..	26

Şekil 3.2: Resim Üzerinde Sıcaklık Haritası Örneği	27
Şekil 3.3: Metin Üzerinde Sabitlemeler ve Sıçramalar.....	28
Şekil 3.4: Reyonadaki Ürünlere Bakış Sırası	35
Şekil 3.5: Web Sayfasındaki Resimlere Ait Odaklanma Oranları.....	35
Şekil 3.6: Göz İzleme Cihazı ile Çevresiyle Etkileşen Engelli Bir Çocuk ve Babası	36
Şekil 3.7: Uçuş Eğitim Simülatöründe Bulunan Bir Göz İzleme Sistemi	38
Şekil 3.8: Sürücü Eğitiminde Kursiyerin Göz Hareketleri Tespiti	38
Şekil 4.1 : Genel bir Beyin Bilgisayar Arayüzü Şeması	39
Şekil 5.1 : Eye Tribe İzleyicinin 3.0 ile USB Bağlantısı	43
Şekil 5.2 : Pritop ile Bilgisayar – Eye Tribe İzleyicinin Mesafe Ayarlanması.....	43
Şekil 5.3 : Eye Tribe İzleyici ile Denekler Arasında Oturma Konumunun Ayarlanması.....	43
Şekil 5.4 : Eye Tribe İzleyici Config Ayar Yapılandırması Sonuç Çıktısı.....	44
Şekil 5.5 : Senkron İşlemlerinde Başarı Durum Gösterimi	44
Şekil 5.6 : Eye Tribe İzleyici Veri Akış Algoritması.....	45
Şekil 5.7 : Eye Tribe İzleyici İlk Ekran Çıktısı.....	46
Şekil 5.8 : Sıcaklık Haritası	47
Şekil 5.9 : Sıcaklık Haritası Akış Şeması	48
Şekil 5.10 : Soru Cevap Algoritması Akış Diagramı.....	49
Şekil 5.11 : Program Ekran Görüntüsü	50
Şekil 5.12 : Soru Database Şeması	57
Şekil 5.13 : Cevap Database Şeması.....	57
Şekil 5.14 : Otomatik Soru Cevap Akış Diagramı.....	58
Şekil 5.15 : Otomatik Soru Cevap Tablo Yapısı.....	62

Şekil 5.16 : Sanal Klavye Ekran Görüntüsü 62

Şekil 5.17 : Sanal Klavye Uygulaması Akış Diagramı..... 63

Şekil 5.18 : Sanal Klavye Database Yapısı..... 65



1. GİRİŞ

İnsan yüzünün ve gözünün belirgin özellikleri, hareketleri; insanın içinde bulunduğu duyguları anlatmakta önemli bir rol oynamaktadır [23]. Göz; insanın dış dünyaya açılan kapısı olan en önemli organlardan biri olduğu için, gözün gerçekleştirdiği görev oldukça önemlidir. Gözler, insan algılamasının yaklaşık yüzde 80'ini sağlamaktadır [83]. Bu yüzden göz takibi 1800'lü yıllardan bu yana popüler bir araştırma konusu olarak günümüze kadar gelmiştir ve son yıllarda birçok alanda kullanılmaktadır. Göz takibi, kullanıcının nereye odaklandığının ve ne kadar süre baktığının incelenmesidir.

Göz takip sistemleri, ilk zamanlarda basit metotlarla kullanıcı hareketlerini kısıtlayan taşınmaz sistemler olarak geliştirilmişlerdir. Son zamanlarda video kameraların görüntü kalitelerinin ve mikroişlemcilerin hızlarının artmasıyla, video tabanlı göz takip yöntemlerinde büyük gelişmeler olmuştur. Elektriksel sinyallerin işlenmesiyle başlayan bu süreç, kameralar yardımıyla gerçekleştirilen bir yapıya dönüşmüştür. İlk zamanlarda kullanıcıya rahatsızlık veren yöntemler, yerini kullanıcı hareketinden bağımsız, uzaktaki kameradan elde edilen görüntülerin işlenmesiyle gerçekleştirilen bilgisayar uygulamalarına bırakmıştır. Uzak sistemlerde kafanın hareketli olması, gözün tespiti ve takibi sırasında yapılması gereken işlemleri artırmıştır. Çünkü kafanın farklı pozisyonlarında yakalanan karelerde, göz içerisinde takip edilen yapılar (göz bebeği, iris vb.) her pozisyon için farklı konumlarda olmaktadır. Bunun için çalışmalarda genellikle 2 Boyutlu (2B) ve 3 Boyutlu (3B) göz modelleri kullanılmaktadır. 2B olarak yapılan çalışmalarda başın pozisyon bilgisine gerek duymadan doğrudan ekran ile göz arasında tanımlanan ilişkiye göre gözün baktığı nokta tespit edilebilmektedir. 3B olarak yapılan çalışmalarda ise başın pozisyon bilgisi kullanılarak uygulamalar gerçekleştirilmektedir.

Göz hareketleri izleme yöntemi; odaklanma sayısı, ortalama odaklanma süresi, toplam odaklanma sayısı gibi verilerden hareket ederek bilişsel süreçlerin gerçek zamanlı olarak ölçülmesinde kullanılabilir [57]. Göz hareketleri verileri; insanların nereye dikkat ettiği, hangi bilgiyi göz ardı ettikleri, en fazla nelerden rahatsız oldukları hakkında bilgi vermektedir. Goldberg ve Kotval'e [19] göre de göz hareketlerini izlemek, kullanıcıların stratejilerini belirleme noktasında önemli bilgiler sunmaktadır.

Son yıllarda; el hareketi, mimikler ve ses ile geliştirilen uygulamalara ek olarak göz hareketi takip sistemleri de birçok alanda problemleri çözmek için insan bilgisayar etkileşimli

uygulamalarda gncel bir konu haline gelmiřtir. Gz hareketi takibi insanların gnlk hayatlarını kolaylařtırmak, eęlence ve tedavi amacıyla tercih edilmektedir. Bu teknolojinin kullanım kolaylıęı sebebiyle ticari olarak pazar payı geniřlemesine raęmen, hala pahalı bir teknoloji olarak kullanıcılara sunulmaktadır. Bunun iin birok arařtırmacı dřk maliyetli sistemler geliřtirmek iin alıřmaktadır.

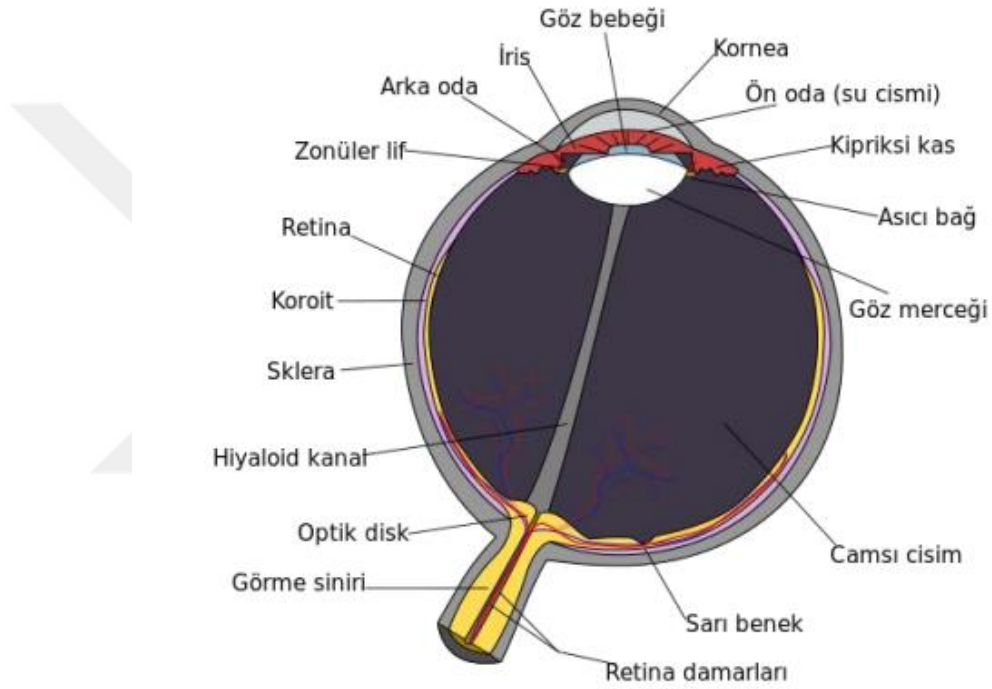
Bu tez alıřmasının 1. Blm kısmında konu hakkında genel bilgiler ve tez alıřmasının amacı verilmektedir. 2. Blmde gz yapısı, hareketi ve gz takip yntemleri; 3. Blmde beyin bilgisayarı arayz iin kullanılan yntemler; 4. Blmde gerekleřtirilen uygulamanın detaylı anlatımı ve sistemin uygulanabilirlik testleri ve sonuları bulunmaktadır.



2. GÖZ HAREKETLERİ TAKİBİ VE BİLİŞSEL SÜREÇLER

2.1. GÖZÜN YAPISI

Göz; görmeyi sağlayan küremsi yarıçapı yaklaşık olarak 12 mm olan bir organdır [79]. Şekil 2.1’de göz yapısı içinde bulunan iris, göz bebeği, kornea, göz merceği, görme siniri, sklera ve retina gibi yapılar görülmektedir. Göz, kafatasında “orbita” denilen kemik bir yuvaya içerisinde. Etraf yumuşak yağ dokusuyla kaplıdır. Göz hareketleri üzerine yapışan 6 adet kas tarafından sağlanmaktadır [83].

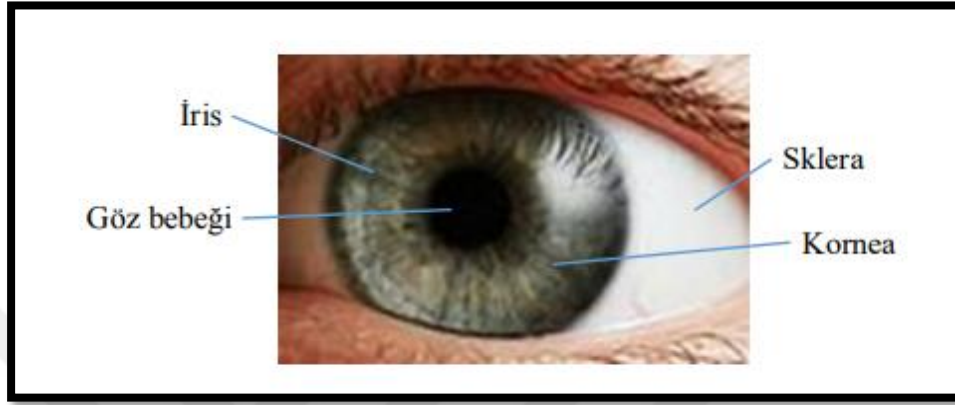


Şekil 2.1 : Gözün Şematik Yapısı

Şekil 2.1’de görülen gözün karşıdan görünüşü üzerinde görüntü işleme uygulamalarında kullanılan temel yapıların görevler şöyledir:

- Göz bebeği; göze belirli miktarda ışık girmesini sağlayan irisin merkezinde çapı değişen bir yapıdır.
- İris; göz bebeğinin etrafında yer alan renkli halkadır.

- Kornea; iris ve göz bebeğini kapsayan şeffaf yapıda gözün odaklama sisteminin bir parçasıdır.
- Sklera; beyaz renkli, üzerini soğan zarına benzeyen kaygan ve damarlı bir mukozanın kapladığı yapıdır.



Şekil 2.2 : Gözün Karşıdan Görünüşü

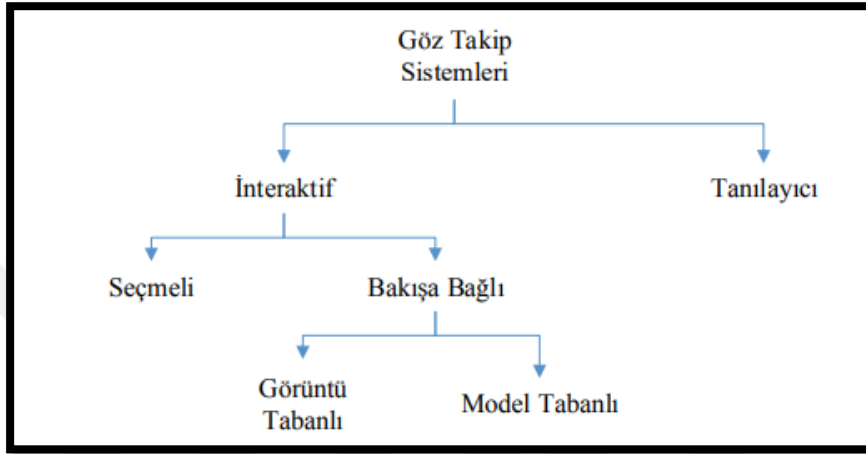
Göz takibinin veriye dönüştürülebilmesi için elde edilen göz hareketlerinin işlenmesi gerekmektedir. Bunun için gözün bazı temel hareketlerinin incelenmesi gerekmektedir. Gözün sürekli olarak yaptığı “sıçrama (saccade)” ve "sabitleme (fixation)" iki hareketi vardır ve gözler uyurken de hareket etmektedir. Sıçrama olarak adlandırılan hareket, iki sabitleme noktası arasında yapılan çok ani kaymadır. Sıçramayı takip eden sabitleme ise, gözün 200 ile 600 ms’lik bir zaman dilimi boyunca durağan kalmasıdır. Aslında göz, bu sabitleme sırasında bile hareketlidir, ancak bu hareketlilik, sıçrama olarak adlandırılmayacak kadar küçük bir hareketliliktir. Bu küçük hareketlilikler, özellikle bakılan cisim de hareketli ise kolaylıkla gözlemlenebilir. Gözün bilgisayar etkileşiminde kullanılmayan pursuit, tremor, rotation, drift gibi başka hareketleri de vardır.

2.2. GÖZ HAREKETİ TAKİBİNDE KULLANILAN YÖNTEMLER

Göz bakışı izleme sistemleri birçok farklı tekniğin birleştirilmesiyle oluşmaktadır ve göz takibi de bunun bir parçasıdır. Genel olarak göz takip sistemlerinin yapısı, algoritmaların benzerlikleri ve farklılıkları şöyledir [78].

Duchowski, göz takip uygulamalarını genel olarak Şekil 2.3’te görüldüğü gibi “interaktif” ve “tanılayıcı” uygulamalar olarak ikiye ayırmaktadır [13]. Tanılayıcı uygulamalar, göz bakış

verilerini kullanıcının görsel ve dikkat süreçlerinin nicel bilgisi olarak kullanmaktadır. İnteraktif uygulamalar ise; göz bakış verilerini, elde edilen göz hareketlerine göre kullanıcı ile etkileşim için kullanmaktadır. İnteraktif uygulamaları “seçmeli” ve “bakışa bağlı” olarak alt kategorilere ayırmaktadır. Seçmeli sistemler göz bakışını fare işaretçisi gibi bir seçme aygıtı olarak kullanırken, bakışa bağlı sistemler karmaşık ekranların (grafik ortamlar vb.) hızlı render edilmesini kolaylaştırmak için kullanıcı bakış bilgisinden faydalanır. Bakışa bağlı uygulamalar da “görüntü tabanlı” ve “model tabanlı” olmak üzere iki gruba ayrılmaktadır.



Şekil 2.3 : Göz Takip Uygulamaları Hiyerarşisi [13]

Göz pozisyonunun ve hareketinin kaydedilmesi metodu “okülografi” olarak adlandırılır. Göz takip sistemlerinde kullanılan ışık tayfı aralığı, kamera sayısı, ölçümleme gerekliliği, kafa hareketinden bağımsız olma gibi farklılıklardan dolayı birçok farklı teknik geliştirilmiştir. Hansen and Ji, bu tekniklerin kullanıldığı uygulamaları aşağıdaki gibi sınıflandırmaktadır [23].

- Kullanıcıya temas eden
- Kullanıcıdan uzak sistemler,
- Öznitelik tabanlı
- Görünüm tabanlı yaklaşım kullanan,
- 2B – 3B bakış tahmin metodu kullanan,

- Kızılötesi
- Gün ışığı kullanan uygulamalar olarak incelenebilir.

1) Kullanıcıya temas eden - Kullanıcıdan uzak sistemler: Kullanıcıya temas eden teknikler, kullanıcıya fiziksel temas eden bazı donanımlar gerektirmektedir. Kontak lens, Elektro-okülografi ve Başa takılan cihazlar kullanıcıya temas eden tekniklerdir. Kullanıcıdan uzak teknikler çoğunlukla görüntü tabanlı olup göz hareketlerini, kullanıcıdan uzak bir kameradan elde etmektedirler. Kullanıcıya temas eden tekniklerin doğruluk oranı kullanıcıdan uzak çalışan tekniklere göre daha fazladır. Fakat bu tekniklerin kullanıcıya verdiği rahatsızlıktan dolayı kullanımı pek yaygınlaşmamıştır.

2) Öznitelik tabanlı – Görünüm tabanlı yaklaşımlar: Öznitelik tabanlı yaklaşımlar, göz görüntüsündeki yansımaları, göz köşesi ve dış hatları gibi yerel özellikleri kullanır. Bakış tahmini için en popüler özellik gözün kızılötesi ışık ile aydınlatılmasıyla elde edilen göz bebeği ve kornea yansımalarıdır. Görünüm tabanlı yaklaşımların, öznitelik tabanlı yaklaşımlara göre gerçekleştirilmesi daha kolay ve gürbüzdür [74]. Görünüm tabanlı metotlar, görüntü içeriğini doğrudan ekran koordinatlarına eşlemeyi kullandığı için ölçümlemeye ve ekran geometrisine ihtiyaç duymazlar [23].

3) 2B eşleştirme – 3B model tabanlı bakış tahmin metodu: 2B eşleştirme tabanlı bakış tahmin metodu için göz bakışı, 3B bakış doğrultusu bilinmeden 2B göz hareketleri seti giriş olarak kullanılarak ölçümlenmiş bakış eşleştirme fonksiyonuyla tahmin edilir. Çıkartılan özellikler genellikle göz bakışıyla değiştiğinden dolayı bunların arasındaki ilişki bakış eşleştirme fonksiyonu olarak kullanılır. Maalesef ölçümlenmiş bakış eşleşme fonksiyonu, kafa hareketlerine karşı çok hassastır. 3B model tabanlı yaklaşım metodu, 3B bakış yönünü hesaplamak için insan gözünün fiziksel yapı özelliklerini modeller. Böylelikle ekranla kesişen bakış noktası basit bir şekilde elde edilir. 3B model tabanlı yaklaşımların çoğu metrik bilgiye dayandığı için ışık kaynağının küresel geometrik modeli, kamera ve ekranın pozisyonu, kamera ölçümlemesi ve bunların uyumunu gerektirmektedir. Fakat bunlar kafa hareketi için düşük duyarlılıktadır [70].

4) Kızılötesi – Gün ışığı kullanan sistemler: Gün ışığı, kızılötesi ışığa doğal bir alternatiftir. Birçok çalışmada kızılötesi ışıktan yararlanılmaktadır. Gün ışığı kullanılan çalışmalarda ışığın

değişmesi, düşük karışıklı resimler gibi problemler olmasına karşın, dış ortamlarda kızılötesi ışığa göre potansiyel olarak daha iyidir [23].

Göz hareketi takibi için kullanılan 4 ana metot aşağıdaki gibi sıralanmıştır [41],[6],[65].:

1. Elektro okülografi

2. Kontak lens

a. Manyetik arama bobin kontak lens (Elektromanyetik)

b. Aynalı kontak lens

3. Kızılötesi okülografi

4. Video okülografi

a. Göz bebeği ve kornea yansımaları

b. Göz bebeği izlemesi

c. İkili purkinje resim kornea yansımaları

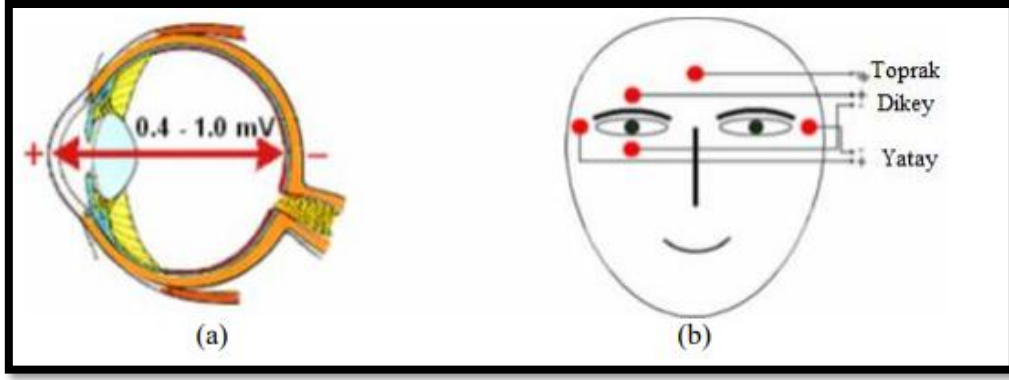
d. Limbus izleme (iris - sklera sınırı izleme)

Görüntü tabanlı yöntemler olarak kategorilere ayrılmıştır.

2.2.1. Elektro Okülografi

Yaklaşık 40 yıldır göz hareketlerini kaydetmek için kullanılan bir yöntemdir [13]. Retinanın yüksek sinir yoğunluğundan dolayı göz içi kutuplanmış yani polarize olmuş durumdadır. Bunun için göz çevresine elektrotlar yerleştirilerek kornea ve retina arasındaki elektriksel farkın ölçülmesinden yararlanılmaktadır. Kullanımı kolay, her iki gözün hareketi kaydedilebilir ve ucuz olması bakımından avantajlıdır.

Bu yöntemin en büyük avantajı gözler kapalıyken de göz hareketlerini tespit edebilmesidir [45]. Fakat gözün yatay ve dikey hareketlerinin tespiti kısıtlıdır. Ayrıca göz pozisyonu belirlenmesi noktasında doğruluğu çok iyi değildir.



Şekil 2.4 : Polarizasyona Göre Bağlantı Şeması a) Göz İçi Polarizasyonu b) Göz Çevresine Elektrotların Yerleştirilmesi [42]

Şekil 2.4.'te bağlantı şeması ve aşağıda yer alan Şekil 2.5'te de yatay ve düşey düzlemde ölçüm alınacak elektrotların kullanıcının göz çevresine yerleştirilmesi görülmektedir. Yıldız çalışmasında göz hareketleri esnasında oluşan elektro okülografi sinyallerinin yazıya çevrilmesi için; kullanılacak karakterlerin gözle çizilmesi yerine daha az sayıda göz hareketi ile kodlanmasını sağlayan bir yöntem geliştirmiştir [83]. Gözlerin sağa, sola, yukarı, aşağı, yukarı sağ çapraz ve yukarı sol çapraz hareketlerini harflerle anlamlandırıp alfabe oluşturmuştur.



Şekil 2.5 : Elektrotların Kullanıcının Göz Çevresine Yerleştirilmesi

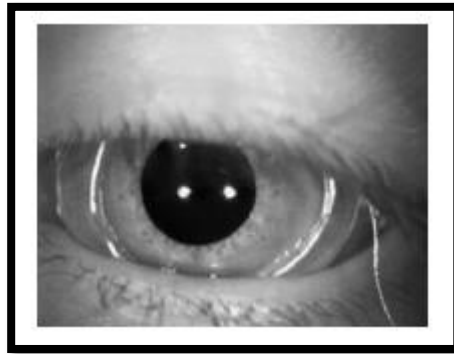
Septanto et al. göz hareketleri ve istemli göz kırpmaları ile kontrol edilebilen bilgisayar faresi benzeri bir cihaz geliştirmişlerdir. Tek kanallı elektro okülograf sinyali cihazın girişi olarak kullanılmıştır. Bir elektrot sağ göz altı ve diğeri sol göz üstüne yerleştirilerek ve zaman alanlı sinyal analizi yapılmıştır [69].

2.2.2. Kontak Lens

Tel bobini manyetik alan içinde hareket ettiği zaman, bobin içinde gerilim indüklenir. Bu bobin göze yerleştirildiği zaman göz hareket sinyali üretilecektir. İnsan göz hareketlerini ölçmek için modifiye edilmiş kontak lens içine küçük bir bobin yerleştirilir. Manyetik arama bobinli göz takip cihazları iki yumuşak kontak lens ve 13 mm çapında tel bobini ile oluşturulur [38]. Kaymaz kontak lens Şekil 2.6'da görülmektedir. Şekil 2.6'da görülen kaymaz kontak lens Şekil 2.7'da görüldüğü gibi göz içine yerleştirilmektedir.

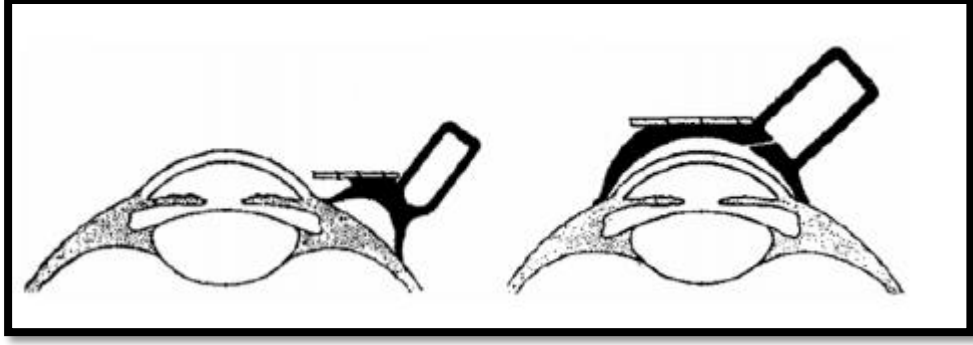


Şekil 2.6 : Manyetik Arama Bobin Kontak Lens



Şekil 2.7 : Sol Göz Üzerine Yerleştirilmiş Kontak Lens

Yansıyan ışığın ölçülmesine olanak sağlayan kontak lens içine yerleştirilmiş aynalı lens Şekil 2.8'de görülmektedir.

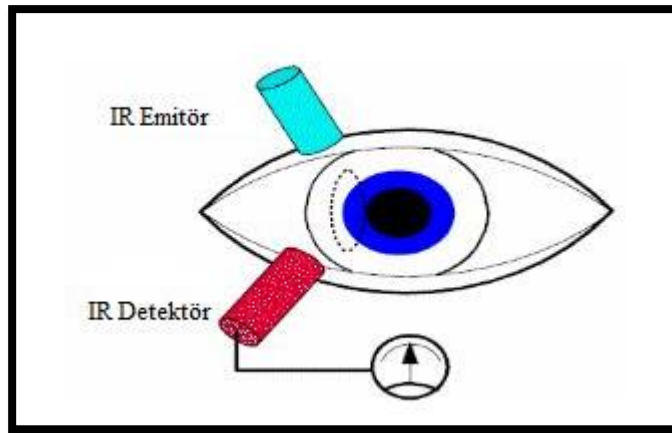


Şekil 2.8: Aynalı Kontak Lensler

Bu yöntem göz hareketlerinin doğallığı hakkında doğru ve hızlı bilgiler vermektedir. Fakat kullanıcılar için lens rahatsız edici olabilir. Bu nedenle insan - bilgisayar etkileşim uygulamalarında yakın zamanda çok tercih edilmemektedir. Daha çok psikolojik ve medikal uygulamalarda kullanılmaktadır.

2.2.3. Kızılötesi Okülografi

Bu yöntemde göz kızılötesi ışık ile aydınlatılır. Kızılötesi okülografi göz içerisinde bulunan skleradan yansıyan ışığın yoğunluğunu ölçer. Göz yüzeyinden geri yansıyan kızılötesi ışık miktarı farkı göz pozisyonu hakkındaki değişim bilgisini verir. Burada kullanılan ışık kaynağı ve sensörler bir gözlük üzerine yerleştirilebilir. Kızılötesi okülografi, elektro okülograftan daha az gürültüye sahiptir, fakat harici ışık değişimlerine daha hassastır. Bu yöntemin temel dezavantajı yatay ekseninde ± 35 derecelik, dikey ekseninde ± 20 derecelik göz hareketini ölçebilmesidir fakat karanlıkta göz hareketlerini ölçebilmesi avantajlarındandır. Şekil 2.9'da temsili çalışma prensibi görülmektedir.

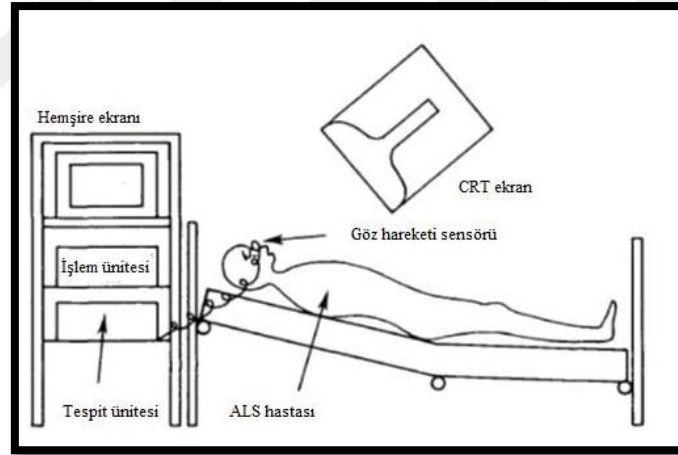


Şekil 2.9 : Kızılötesi Okülografi Metodu Çalışma Şekli

Kızılötesi okülografi, görüntü işleme uygulamalarının kullanıldığı kornea yansıması, Purkinje resim ve göz bebeği izlenmesi teknikleri kullananan birçok ticari uygulamada tercih edilmektedir [5].

Reulen et al. çalışmalarında limbus izleme tekniğinin bir çeşidi olan kızılötesi okülografiyi kullanmışlardır. Gözün üstüne ve altına sırasıyla kızılötesi ışık yayan diyotlar ve kızılötesine duyarlı foto-transistörler yerleştirmişlerdir. Foto-transistör yansıyan kızılötesi ışığı gerilime dönüştürür. Altta ve üstte bulunan foto transistörlerin gerilim farkı gözün açılma sapmasıyla orantılıdır [59].

Yamada and Fukuda, hastaların göz hareketleriyle kontrol edebilecekleri “göz kelime işlemci” çalışmasını gerçekleştirmişlerdir. Göz hareketlerini tespit edebilmek için gözün karşısına foto-transistör ve kızılötesi led yerleştirmişlerdir. Çalışmada tek gözün yatay hareketlerini incelemiştir. Dakikada sekiz kelime girişine ve hemşire çağırma fonksiyonuna sahip olan sistemin kurulumu beş dakika almaktadır ve hemşirelerce kullanım kolaylığı onaylanmıştır [81]. Sistemin genel yapısı Şekil 2.10’da görülmektedir.

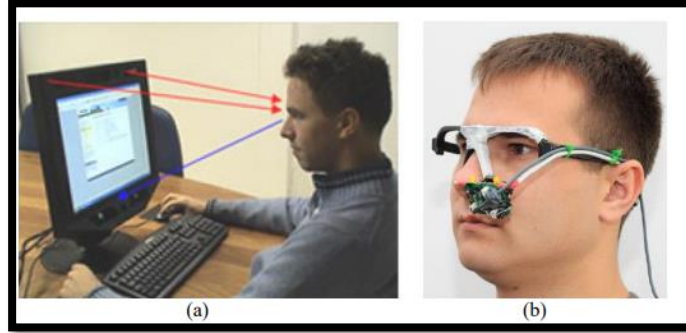


Şekil 2.10 : Hasta ve Göz Kelime İşlemci Düzenegi

2.2.4. Video Okülografi

Video okülografi, ticari yazılımlar tarafından son zamanlarda göz hareketi takibi için en çok tercih edilen yöntemdir. Ayrıca teknolojinin gelişmesiyle birlikte pahalı ve karmaşık uygulamalar, yerini daha kullanıcı dostu ve uygun fiyatlı uygulamalara bırakmaktadır. Göz görüntüleri bir veya daha fazla kamera ile elde edilebilir. Bu sistemlerde genellikle basit ya da karmaşık düzeyde ölçümler bir ya da birden fazla kamera ile yapılabilir. Video okülografi tabanlı teknikler Şekil 2.11’de görüldüğü gibi kullanıcıya temas eden, etmeyen ya da kafaya

monte edilebilen sistemler olabilir. Gün ışığı ya da kızılötesi ışık çalışmanın durumuna göre tercih edilmektedir [22]. Temassız yöntemler, insan bilgisayar etkileşimli uygulamalarda daha çok tercih edilmektedir [25].



Şekil 2.11 : Video Ökülografi Tabanlı Teknikler (a) Uzak Göz Takip Cihazı (b) Kafaya Monte Göz Takip Cihazı Konumu [41]

a. Limbus izleme (iris-sklera sınırı izleme): Limbus, beyaz renkli sklera ile koyu iris çizgisi arasında kalan sınırdır. Bu bölgeler arasında oluşan karşıtlık farkından dolayı aradaki sınır, ışık altında kolayca tespit edilir ve izlenir. Bu teknik limbusun pozisyonu ve şekline bağlı olduğu için kafanın sabit olması veya aparatların kafaya takılması gerekmektedir. Genelde kızılötesi ışık kullanılmaktadır. Göz kapaklarının limbusu alttan ve üstten çevrelemesinden dolayı limbus izleme sadece yatay izleme için uygundur [68]. Şekil 2.12’de görülen aparat göz bebeği ve limbus izlemek için uygun fiyatlı olarak yaklaşık 700\$ maliyet ile geliştirilmiştir.

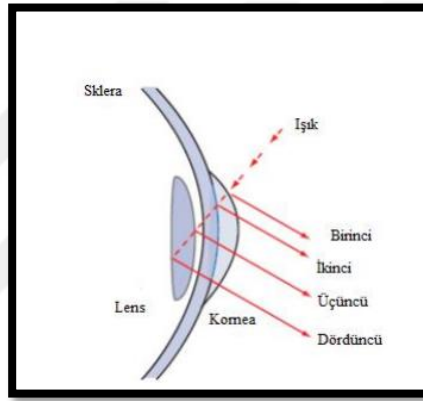


Şekil 2.12 : Göz bebeği ve Limbus İzlemek İçin Geliştirilen kafaya Monte Gözlük Tasarımı [60]

b. Göz bebeği izlemesi: Göz bakışı doğrultusunu izlemek için kullanılan bu teknik limbus izleme tekniğine benzemektedir ve kafanın hareketinden bu teknikte etkilenmektedir. Farklı olarak izleme için burada göz bebeği ile iris arasındaki daha küçük olan sınır kullanılmaktadır. Göz bebeği tespiti ve izlemesi iris ile olan karşıtlık farkının küçük

olmasından dolayı daha zordur. Birçok çalışmada bu olumsuzluğu aşmak için kızılötesi ışık [75]. Limbus izleme tekniğinden avantajlı olan yönlerinden biri; limbusa göre göz kapakları tarafından daha az çevrelendiği için dikey hareketlerin izlemesinde gerçekleştirilebilmektedir. Diğer ise; göz izleme doğruluk oranı daha iyidir.

c. Göz bebeği yansıması: Bu yöntemle yapılan çalışmalarda, sistemin karşısında oturan kullanıcının gözüne bir veya birden fazla kızılötesi ışık gönderilerek göz üzerine düşmesi sağlanır. Tek bir ışık kaynağı göz üzerindeki farklı katmanlarda birden fazla yansıma oluşturmaktadır. Şekil 2.13.'te görülen göz üzerine düşen ışığın kornea ve lens arasındaki sınırdaki oluşan yansımalarına Purkinje resim denilmektedir [12]. Şekil 2.13'te görülen ilk Purkinje resim ya da kornea yansıması "parıltı" olarak adlandırılır [23].

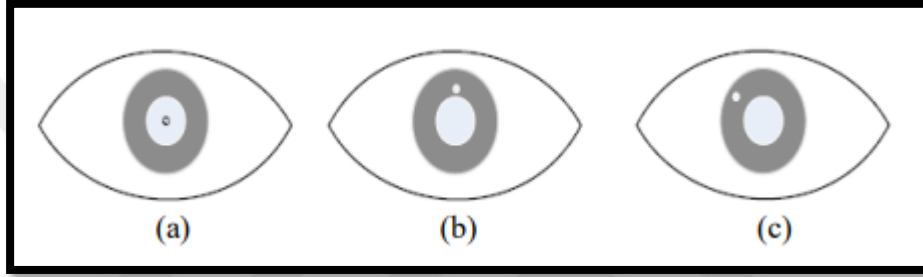


Şekil 2.13 : Işığın Gözden Yansıması Sonucu Oluşan Birinci, İkinci, Üçüncü ve Dördüncü Purkinje Resimler.

Göz üzerinden yansıyan ışık Şekil 2.14'te görüldüğü gibi bir parıltı oluştururken aynı zamanda göz bebeğinin koyu bir disk şeklinde görünmesini sağlar [87]. Şekil 2.15'de gözün farklı bakış durumlarında parıltıların yerleri görülmektedir.



Şekil 2.14 : Yansıyan Işığın Etkisi (a) Işık Kaynağının Göz Üzerinde Oluşturduğu Parıltı (b) Kornea Yansıması.

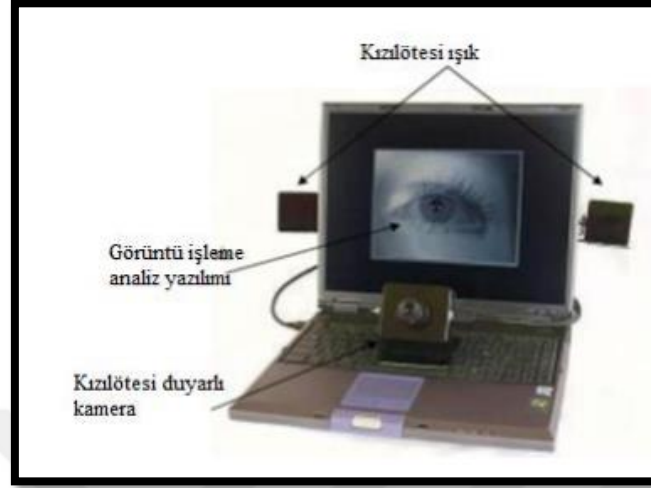


Şekil 2.15 : Parıltı yerleri (a) Göz ışık kaynağına doğru bakarken (b) Göz ışık kaynağının alt kısmına bakarken (c) Göz ışık kaynağının sol alt kısmına bakarken.

Bu yöntem kullanılarak parıltı ve göz bebeği arasındaki vektörel ilişkiyle gözün nereye baktığını tespit eden birçok çalışma literatürde yer almaktadır [48]. Bu çalışmalar, genellikle kafanın sabit kalmasını sağlayacak bir donanıma ihtiyaç duymaktadır. Göz bebeği ve parıltı arasındaki vektör ile ekran koordinatları arasında eşleştirme yapabilecek farklı dereceden tanımlanan fonksiyonların parametreleri o andaki kafa pozisyonuna göre hesaplanmaktadır. Farklı baş pozisyonları için tekrar ölçümlene yapılması ve kafanın sabit tutulması zorunluluğunun kullanıcıları rahatsız etmesi, serbest kafa pozisyonunda göz bakış doğrultusunun kestirimini yapabilen sistemlerin gereğini ortaya çıkarmıştır. Sabit kafa pozisyonu problemine kornea yüzeyini düz kabul ederek çözüm üreten çalışmalar da literatürde yer almaktadır [55].

Kafa pozisyonunu tahmin etmek için 2 tane kızılötesi ışık kaynağı kullanılan bu sistemlerde, eğitim aşamasında çeşitli kafa pozisyonlarında bakılan düzlem ile düz kabul edilen kornea düzlemindeki göz bebeğinin parıltıya göre konumu arasında ilişki kurulmuştur. Fakat gözün kenarlara doğru gitmesi durumunda, kornea yüzeyinin düzlemsel kabul edilmesi varsayımının

geçerliliğinin azalması bu sistemlerin en önemli dezavantajıdır. Göz bakış doğrultusunu, yapay sinir ağları yöntemiyle belli bir açı aralığına kadar kafanın oynamasına izin verecek şekilde çözen çalışmalar da vardır [88].



Şekil 2.16 : Göz Bebeği ve Kornea Yansıması Sistemi [6]

Şekil 2.16’da temassız, 2 tane kızılötesi ışık kaynağı kullanılan bir göz bebeği ve kornea yansıma sistemi görülmektedir.

d. İkili Purkinje resim kornea yansıması: Göz yuvarlağının kendi etrafında dönmesine karşın, Purkinje resim nispeten daha sabittir. Birinci ve dördüncü Purkinje yansımalarının ölçülmesiyle, gözün ötelenme ve dönme hareketi göz takipçisi tarafından ayrılır. Her iki yansıma göz ötelenmesi boyunca aynı şekilde hareket eder. Fakat görüntüler dönme boyunca farklı miktarda hareket eder. Maalesef ikili Purkinje resim kornea yansıması kullanan göz takip ediciler oldukça hassas olmalarına rağmen, kafanın sabit olmasını gerektirebilirler. Bu tekniğin doğruluğu genellikle yüksektir, fakat dördüncü Purkinje görüntüsü zayıf olduğundan ışıklandırma koşullarının ciddi şekilde kontrol edilmesini gerektirmektedir.

e. Görüntü tabanlı yöntemler: Bu yöntemlerde ilk olarak bir veya birden fazla kamera ile görüntü üzerinden göz resmi elde edilir. Yapay ışık ve kızılötesi ışık kullanılarak göz bölgesi, kafa pozisyonuna göre öznitelik ve görünüm tabanlı metotlarla göz bakış tahmini gerçekleştirilir. Öznitelik tabanlı yaklaşımlar; bakış tahmini için kullanılan limbus ve göz bebeği hatlarını, göz köşeleri ve kornea yansıması gibi gözün ayırıcı özelliklerini tanımlayarak gözün karakteristiğini ortaya çıkarır. Bu yaklaşımın amacı; gözün bilgi verici yerel özelliklerini tanımlamaktır. Bu özellikler genel olarak ışık değişimi ve bakış açısı değişimlerine daha az duyarlıdır [27].

Fakat bu sistemlerin dış ortamda ve güçlü ışık altında performans problemleri vardır. Ayrıca iris ve göz bebeği özellikleri doğru tespit edilemediği zaman bakış tahmin doğruluğu düşer. Öznitelik tabanlı yaklaşımların model ve interpolasyon tabanlı olmak üzere iki tipi vardır [23]. Model tabanlı yaklaşımlar, 3B bakış yönü vektörünü tahmin etmek için gözün geometrik modelini kullanır. Sinir ağlarındaki gibi polinomal veya parametrik olmayan formda belirli bir parametrik forma sahip olan bu yöntemler görüntü özelliklerinden bakış koordinatlarını (2B veya 3B) eşleştirme varsayımına göre çalışır. İnterpolasyon tabanlı yöntemler açıkça insan gözün geometrisini ve fizyolojisi modellemekten kaçınırken, bunun yerine bakılan noktayı görüntü özelliğinin genel bir fonksiyonu olarak tanımlar [5].

Görüntü tabanlı metotlar; gözü direkt fotometrik görünüşüne göre tespit eder ve izler. Bu yöntem öznitelik tabanlı yaklaşıma göre daha gürbüz olmasına rağmen kafa hareketlerinden etkilenmesi yaklaşımın olumsuz yönünü oluşturmaktadır [70].

2.2.5. Göz Hareketi Takibinde Kullanılan Tekniklerin Karşılaştırılması

Tablo 2.1’de geleneksel göz takip cihazlarının karakteristik özelliklerinin karşılaştırılması görülmektedir. Bu çizelge ile yöntemlerin biririne göre uygulanabilirlik durumları kolayca tespit edilebilir [47].

Tablo 2.1 : Göz Hareketi İzleme Teknolojilerinin Karşılaştırılması [47]

Kullanılan Teknoloji	İzleme Tekniği	İzleme Doğruluğu	Örnekleme oranı	Yorum
Kontak lens	Göz içine yerleştirilmiş kontak lensten yansıyan ışığın izlenmesi	< 0.1°	>100 Hz	Çok zorlayıcı fakat hızlı ve doğru
Elektro okülografi	Göz çevresindeki elektropotansiyelin ölçülmesi ile	0.1°-0.5°	>100 Hz	Zorlayıcı fakat basit ve düşük maliyetli
İkili Purkinje resim kornea yansımaları	Kornea ve lens sınırından ışık yansımalarının video izlenmesi	0.1°-0.5°	>100 Hz	Zorlayıcı değil, kafanın durumu önemli
Limbus izleme	İris- sklera sınırının video izlenmesi	>0.5°	>100 Hz	Kamera tabanlı, düşük dikey doğruluk
Göz bebeği izleme	Koyu renkli göz bebeğinin video izlenmesi	0.1°-0.5°	>50/60 - 100 Hz	Kamera tabanlı, göz bebeği tespiti zor
Göz bebeği	Kornea ve koyu		>50/60 -	Kamera tabanlı,

ve kornea yansımaları	renkli göz bebeğinden yansıyan ışığın video izlenmesi	< 0.1°	100 Hz	bazı kafa hareketleri tolere edilebilir
Manyetik arama bobin kontak lens	Göz içindeki bobinin elektromanyetik izlenmesi	< 0.1°	>100 Hz	Çok zorlayıcı, hızlı ve doğru bilgi verir
Kızılötesi okülografi		2°		Kafaya monte, limbus izleme
Görüntü tabanlı	Bilgi tabanlı, değişmez öznelik tabanlı ve şablon eşleştirme tabanlı yöntemlerle izleme	0.5°-2°		Kamera tabanlı, ölçümleme gerekli

2.3. GÖZ HAREKETLERİ VE BİLİŞSEL SÜREÇLER

2.3.1. Dikkat

Bilişsel psikoloji alanında yapılan çalışmalar, dikkatin bir anda birden fazla nesneye odaklanamadığını göstermektedir. Duyu organlarından gelen zengin öğelerin hepsinin birden bilişsel sistem tarafından ele alınması mümkün değildir. Dikkat sayesinde gerekli olan bilgiler seçilir ve gereksiz olanlar sistemi boşuna işgal etmemesi için elenir [16]. Dikkat; aynı anda beliren nesne veya olaylardan bir tanesini, zihnin, bilinçli veya istekli olarak ön plana almasıdır. Seçici dikkat ise bir uzamsal-mekânsal alana bakıldığında önemli bulunan bir öğeye odaklanılmasını, diğer öğelerin geri planda tutulması veya alan dışında bırakılmasını savunur. Buna göre iki yaklaşım vardır [73]:

Filtre Modeli (Broadbend): Gelen uyarıcılarda dikkati çekene odaklanıldığını ve seçici filtrede diğer kanalların kapatılarak sadece bu sinyalin beyne iletildiğini savunur. Yani odaklanılan nesne dışındaki öğelerin farkında olunmaz.

Zayıflatma Modeli (Treisman): Bu model, Broadbend'in tek kanal modeline karşı çıkmaktadır. Çünkü bir nesneye odaklanıldığında da çevrede olanları duyarız. Ancak seçici filtre bu noktada diğer sinyalleri engellemez, sadece zayıflatır [73].

Çok dikkat çeken nesnelere; 50 ms.'de fark edilmektedir. Buna "göze çarpma etkisi" denir. Treisman'a göre bu süreç iki aşamada gerçekleşir. İlk aşama dikkat öncesi aşamadır ve ortam taranır. İkinci aşama ise nesnelere fiziksel özelliklerine göre ayırt edilerek bilişsel haritalara kodlandığı aşamadır [73]. Soruların tasarımında göze ilk çarpan unsur, ilk odaklanma süresi ve noktası ile belirlenebilir.

Dikkat ve seçici dikkat ile ilgili kuramlara göre sorunun farklı biçimlerinde ilk odaklanma süresi ve ilk odaklanma noktası değişiklik gösterebilir. Soru tasarımında önemli unsurlara dikkat çekmek için renk, boyut, yön, hareket gibi farklılıklara yer verilmelidir. Ancak çok fazla veya gereksiz öğelere yer vermek, dikkatin dağılmasına neden olabilir. Özellikle olumsuz ekli sorularda (değildir, bulunmaz...) kelimenin altının çizilmesi ile dikkat çekmek amaçlanmaktadır.

Dikkat; genel anlamıyla bir sistemin girdilerini seçici olarak süzgeçleyerek işlem yükünü azaltan mekanizmalar olarak tanımlanır ve gözlere gelen yoğun bilgi akışı da insan beyinde pek çok seviyede süzgeçlenerek görsel dikkat sistemini oluşturur. Beyinde bilgi işlemenin

fiziksel kaynak kullanımını açısından (ör: oksijen kullanımı veya nöron tahsisi) sınırlı olması böyle bir sisteme ihtiyaç duyulmasının temel sebebi olarak görülebilir. Dikkat mekanizmasının görsel bilgiyi süzgeçlemesindeki seçim süreci bilgiyi şekillendirdiği için bilgiyi daha etkin şekilde temsil eder ve böylece daha yüksek bilişsel işlevlerin işlerini kolaylaştırır [64].

Görsel dikkat mekanizması;

- Bir ilgi alanı bölgesi seçilmesi
- İlgilenilen değer ve özelliklerin seçilmesi
- Görsel sistemi oluşturan nöron ağlarındaki bilgiyi kontrol etme
- Zaman içinde seçili bir bölgeden diğerine geçebilme gibi temel bileşenler içerir [76].

2.3.2. Algı

Algı; kısaca duyumsadıklarımızın yorumlanması olarak tanımlanabilir. Algılar; önyarıntılar, önyarıntılar, geçmişte kurduğumuz hipotezlerden etkilenebilir. Algısal uzam ise kısa bir zamanda edindiğimiz deneyimler olarak tanımlanır. Algısal uzam ile ilgili çalışmalar, göz ile başlamıştır. Çünkü göz; diğer duylulara göre daha kolay çalışma yapmaya olanak verir [73]. Tasarımda algısal uzam; kullanıcıların birim zamanda ekrandan elde ettikleri bilgidir. Algısal uzamın okuma çalışmalarında bir seferde 4-5 kelime ile sınırlı olduğu düşünüldüğünde ekrandaki metin boyutları, soru tasarımındaki nesnelerin birbiri arasında bırakılan mesafelerin, grafiklerdeki öğelerin boyutlarının ve konumlarının belirlenmesinde önemli bir etkisi vardır. Özellikle ALES sınavında şekil yeteneği alanında sorulan sorularda algı büyük önem taşımaktadır. Bu tip soruların tasarımlarındaki farklılıkların soruların algılanmasına, başarıma ve cevaplama süresine etkilerinin belirlenmesi gerekmektedir.

Mayer [44], Türetimci Çoklu Öğrenme Kuramında (Generative Multimedia Learning Theory); bir materyalin farklı formlarda (ses, resim, hareket) sunumunu ele almış ve farklı bilişsel kuramlardan yararlanarak öğrenme ortamlarındaki sunumlara açıklama getirmiştir. Mayer'in Görsel Bitişiklik İlkesine göre (Spatial Contiguity Principle); görsellerle ilgili açıklama yazmak gerektiğinde, resim ile metni ayrı yerde vermek yerine vurgu yapılmak istenen yere kelimelerin yazılması daha uygundur. Gözün yorulmaması ve zihinde ilişkilendirmenin daha rahat yapılabilmesi açısından ekrana yerleştirilen görseller ve metinler düzgün hizalanmalı ve

ilişkilendirilmelidir. Dikkatin dağılmasını önlemek için içerik tam ekranlardan oluşacak şekilde tasarlanmalıdır.

Bazı sorularda tablo-grafik yorumlamayı gerektiren maddeler bulunmaktadır. Bazen bu sorular sadece tablo içermekte, bazen de grafik verilerek sorulara cevap verilmesi istenmektedir. Bireylerin sahip olduğu sınırlı kapasite düşünüldüğünde; soruların grafik, tablo veya sadece metin olarak sunulması durumunda göz hareketlerinde ne gibi değişikliklerin olacağı önemli bir sorudur. İkili Kodlama kuramına göre; görsel ve işitsel bilgiler ayrı bilgi işlem kanalları vasıtası ile işlenirler. Her bir bilgi işlem kanalı sınırlı kapasiteye sahiptir ve bilginin farklı kanallarda işlenmesi uygun zihinsel modelleri yapılandırmak için tasarlanmış aktif bilişsel bir süreçtir [2].

2.3.3. Bağlam Etkisi

Soru tasarımında bir diğer önemli unsur da bağlam etkisidir. Geon Teoriye göre bir öğenin tanınması kendi bağlamı içinde olduğunda daha kolaydır [73].

2.3.4. Örüntü Tanıma

Görsel örüntü tanıma süreci üç farklı hipotezi öne sürer. Gestalt psikologları; örüntü tanıma sürecini yakınlık, benzerlik ve kendiliğinden tamamlama ilkelerine dayalı tanıma olarak tanımlar. Eğer önce görüntüyü oluşturan parçalardan hareketle bütün tanınıyorsa aşağıdan yukarı tanıma söz konusudur. Yukarıdan aşağı tanımada ise önce bütün, sonra parçalar tanınır. Şablon eşleme teorisi ise bir nesne ile karşılaşıldığında onun tanınmasının zihindeki şablonların karşılaştırılması ile mümkün olduğunu savunur [73].

Eğer soru tasarımında tanınması zor olan veya sınavı alanların şablonlarıyla eşlenemeyen öğeler varsa bunların tekrar düzenlenmesi gerekebilir. Ayrıca bu sorularda zemin-figürün renkli veya renksiz olması ya da taralı - boyalı alan içerip içermemesi; ilgili örüntülerin tanıma sürecinde fark yaratabilir.

2.3.5. Zihinsel İmgeleme

O anda var olmayan bir nesne veya olayın zihinde canlandırılmasıdır. Zihinsel imgelemenin nasıl oluştuğuyla ilgili üç farklı yaklaşım vardır. Bunlardan ikili kodlama hipotezi; nesnelere zihinde görsel veya kavramsal olarak tutulduğunu söyler. Kavramsal-önermesel hipotez;

nesnelerin zihinde soyut kavramlar olarak tutulduğu ve ilişkilendirildiğini söyler. İşlevsel-eşitlik hipotezi ise nesne veya kavramların işlevlerinin zihinde tutulduğunu öne sürer [73].

Zihinsel imgelemenin, bireylerin günlük hayatlarında; bellekten geri çağırma, tanımlar üretme, zihinsel uygulama, motivasyonel durumlar, düşünme ve ilişkilendirmenin yanı sıra problem çözme süreçlerinde de kullanıldığı ortaya konmuştur. Bu noktadan hareketle ALES şekil yeteneği ve bağlam sorularındaki problem çözme süreçlerinde zihinsel imgelemenin var olduğu ileri sürülebilir.

2.4. GÖZ HAREKETİ VERİLERİ

Göz hareketlerini izleme, göz hareketlerinin ölçülerek herhangi bir anda nereye ne sıklıkla baktıklarını ve gözlerini bir bölgeden başka bölgeye yönelttiklerini inceleyen bir tekniktir. Bireylerin göz hareketlerinin incelenmesi, İnsan-Bilgisayar etkileşimi üzerinde çalışan araştırmacıların görsel bilgi işleme süreçleri ve sistem ara yüzlerinin kullanılabilirliğine etki eden etmenler üzerine çalışmalarını sağlamaktadır. Ayrıca göz hareketleri engelli bireylerin fare ya da klavye kullanmadan sadece göz hareketlerini bir giriş birimi gibi kullanmasını da sağlayabilmektedir [53]. Bu bağlamda; odaklanma, sıçrama ve izlenen yol olmak üzere üç ana veri ve bu verilerin alt türevleri, bilişsel süreçler hakkında bilgi edinmekte önem kazanmaktadır.

2.4.1. Odaklanma (Fixation)

Odaklanma noktası, o anda hangi verinin işlendiğini belirtir [35]. Odaklanmalar; araştırmanın yapıldığı bağlama bağlı olarak farklı şekillerde yorumlanabilir. Bir kodlama görevinde (encoding task) belirli bir bölgede toplanan sık odaklanmalar; ilgili bölgenin görevi tamamlamak için önemli olduğu anlamına gelebilir. Ayrıca ilgili bölgenin kodlanmasında güçlük yaşandığı yönünde de yorumlanabilir. Bir arama görevinde ise bu veriler başka anlamlarda yorumlanabilir. Yüksek sayıdaki tekil odaklanmalar veya odaklanma kümeleri (clusters of fixations) nesnenin farkına varılma sürecindeki tereddütleri işaret eder [26]. Goldberg ve Kotval'e [19] göre çok uzun süren odaklanmalar; daha kısa süren odaklanma süreleri kadar anlamlı olmamaktadır.

Odaklanma Sayısı (Number of Fixations): Ekrandaki nesnelere yapılan odaklanma sayısını verir. Odaklanma sayılarının çok fazla ve ekrana yayılmış olması kullanıcının “kafasının karıştığı” şeklinde yorumlanabilir. Arama görevinde odaklanma sayısının artması, etkisiz bir arama süreci geçirildiğini gösterir [19].

Odaklanma Süresi (Fixation Duration): Ekrandaki ögenin ne kadar süre boyunca işlendiğini (process) milisaniye (ms) olarak verir. Odaklanma süresinin uzunluğu, o nesnenin işlenmesinde yaşanan zorluğu gösterir [35].

Her Bir İlgi Alanına Yapılan Odaklanmaların Sayısı (Number of Fixations Per Area of Interest): İlgi alanları araştırmacı tarafından belirlenen alanlardır. Genellikle menüler, bağlantılar, önemli resim veya tablolar ilgi alanı olarak belirlenir. Bu alana yapılan odaklanma sayısı, bu alanlara “kullanıcının verdiği önemi” belirtir [53].

Tekrarlı Odaklanmalar (Repeat Fixations): Belirli bir nesneye tekrarlı yapılan odaklanma, ilgili nesnenin “anlamlandıramadığı” ya da görselin “görülemediği”, “bulanık olduğu” anlamına gelebilir [19].

İlk Odaklanma Zamanı (First Fixation Time): Tasarımda ilk dikkati çeken unsuru belirlemeyi sağlar. Eğer çok önemli bir unsur ilk dikkati çekmiyorsa, kullanılabilirlik açısından bir problem olduğuna yorumlanabilir.

Bakış Süresi (Gaze Duration): Bir noktaya yapılan odaklanma sürelerinin toplamıdır. İlgili nesneyi kullanıcının anlamlandırmada zorluk yaşadığı veya bağlam açısından yanlış bir noktada bulunduğu anlamına gelebilir.

2.4.2. Sıçrama (Saccade)

Sıçramalar; görsel alanın istenilen kısmını fovea üzerine getirmek için göz yuvarlağının yönlendirilmesi ile gerçekleşen hızlı göz hareketleridir. Bu hareketler hem istemli hem de refleks olabilir. Gözün sıçrama hareketleri 10 ms'den 100 ms'ye kadar değişen aralıkta gerçekleşebilir. Göz sıçramaları balistik ve stereotip olarak düşünülür. Stereotip; belirli hareket örüntülerinin tekrarlı bir şekilde gerçekleşmesidir.

Balistik sıçramada ise göz sıçramasının hedef noktası önceden programlanmıştır. Balistik göz sıçramaları (saccade) başlamadan önce hedefin belirlenmiş olması gerekmekte, diğer yandan hedefin genellikle foveanın dışında olması nedeniyle bu hedef düşük netlikteki periferel görüşle seçilmelidir [12].

Sıçramalar genellikle dikkatin odak noktasının istemli bir şekilde değiştirildiğinin göstergesi olarak kabul edilir. Sıçramalar süresince kodlama yapılmaz. Bu sebeple nesnenin tanınmasıyla ilgili bir bilgi verilmez. Ancak tekrarlı yapılan sıçramalar ve geri dönüşler, nesnenin

tanınmasında yaşanan güçlüğü belirtebilir [57]. Okuma sürecinde yapılan sıçramalar oldukça küçüktür. Genellikle iki veya üç harf boyunca yapılır. Sıçrama sayısının artması, metnin kavranmasında yaşanan güçlükleri gösterebilir.

Sıçramaların Sayısı (Number of Saccades): Sıçrama sayısının fazla olması, tasarımın iyi olmamasına veya verimsiz bir arama sürecine işaret edebilir. Sıçrama Genişliği (Saccade Amplitude); sıçramanın aniden ve uzaktaki bir nesneye olması, o nesnenin sonradan belirdiğini ve aniden dikkati çektiğini gösterir. Geriye Yönelik Sıçramalar (Regressive Saccades) ise bir nesnenin daha az anlamlandırıldığı anlamına gelir [19].

2.4.3. İzlenen Yol (Scanpath)

İzlenen Yol (Scanpath): İzlenen yol; odaklanma-sıçrama-odaklanma sırasıyla hareket eden bir süreci gösterir. Bir arama sürecinde optimum izlenecek yol hedefe olan doğrudan çizgi ile gösterilebilir [19].

İzlenen Yol Süresi (Scanpath Duration): İzlenen yol süresinin artması bir tarama (scanning) sürecinin verimsiz olduğu anlamına gelebilir [19].

İzlenen Yol Uzunluğu (Scanpath Length): Ekranda gözün çizdiği yollar çok uzunsa (sürekli gidiş-geliş ve zigzag hali) kullanıcının “kafasının karıştığı” veya “aradığını bulamadığı” anlamına gelebilir. Ekranda çok fazla dikkat çekici öğenin bulunması da gözün ekranda çok dolaşmasına neden olabilir. Ekrandaki en verimli arama, izlenen yolun kısa olması ile belirlenebilir [19].

Geçiş Matrisi (Transition Matrix): Bir alandan diğerine geçişleri dikkate alarak arama sırasını gösterir. Alanlar arasındaki geri dönüşleri ve ileri gidişleri gösterir [19].

İzlenen Yolun Düzensizliği (Scanpath Regularity): “döngüsel tarama davranışı” olarak tanımlanan ve “normallikten” sapma gösteren davranışları belirtir. Bunun nedeni genellikle kullanıcıların deneyiminin azlığından veya kötü arayüz tasarımından kaynaklı arama problemlerinden meydana gelir [19].

Sıçrama / Odaklanma Oranı (Saccade / Fixation Ratio): Aramaya harcanan vaktin (saccade), veri işlemeye (fixation) harcanan süreye oranlanmasıdır. Oranın azalması, daha çok işlemenin ve daha az aramanın olduğu anlamına gelir [19].

2.4.4. Akıcı İzleme (Smooth Pursuit)

Gözün akıcı hareketi genellikle görsel alandaki hareketli bir nesneye tepki sürecinde gerçekleşir. Akıcı izlemenin amacı retina üzerinde oluşan hareketli nesnenin imgesini ayrıntılı bir şekilde algılayabilmek için sabitleştirmesidir. Böylece hem hedef imgenin en yüksek görseli keskinlikle retina üzerinde konumlandırılması hem de imgenin hızının azaltılmasıyla doğru ve titiz bir şekilde hedefin izlenmesi, bu hedefin uzamsal biçimin algılanmasını kolaylaştırır. Akıcı izleme hareketi (smooth pursuit), hareketli nesnelere takip eder ve sıçramalardan oldukça yavaştır. Statik bir ekranın gözlemlenmesi (bilgisayar ekranı) sırasında her ne kadar gözün hareket ettiği hissi oluşsa da hareketli bir uyaran olmadığı sürece akıcı izleme gerçekleşmez [32]. Hedef hareketin (motion) sınırına bağlı olarak, gözler hedefle beraber aynı hızda hareket edebilme kapasitesine sahiptir. Takip hareketleri negatif geri bildirimdeki kontrol sistemlerine örnek olabilir [13].

2.4.5. Görsel Veriler

Bakış Grafiği (Gaze Plot): Bakış noktaları, odaklanmalar, izlenen yolları sırası ile birlikte bir veya birden fazla kayıt üzerinde gösteren araçtır (Tobii: Bir eye-tracker markasıdır). Farklı bireylerin aynı madde üzerindeki ortak davranışları ve aynı bireyin benzer maddelerdeki farklı davranışları ortaya konabilir.

Sıcaklık Haritası (Heat Map): Bir veya çoklu kayıt üzerindeki bakış verilerinin toplamından oluşan bir sıcaklık haritası aracıdır (Tobii). Katılımcıların yoğun olarak baktığı bölgeler; bakma sayısı ve süresine bağlı olarak yeşil-sarı ve kırmızı olarak renklendirilmektedir. Böylece şekli yeteneği ve bağlam sorularındaki sunum farklılıklarında değişiklik gösteren göz hareketleri belirlenebilir.

Kümeleme (Cluster): Katılımcıların sıklıkla odaklanma yaptığı ilgi alanlarını görselleştirir. Doğrudan bakış yapılan bölgeleri ilgi alanı olarak belirler (Tobii: bir eye-tracker markasıdır.). Soru maddelerinde bireylerin yoğun odaklandığı bölgeler belirlenebilir.

İlgi Alanı (Area of Interest – AOI): Araştırmacıların ilgi alanını belirleyebildiği ve sadece bu alanlar üzerindeki göz hareketi verilerini elde edebildiği araçtır (Tobii). İlgi alanı aracılığı ile bağlamdaki kavramlar, beceriler, sorular ve seçenekler, şekillerin iç ve dış kısımları ilgi alanları olarak belirlenecektir. Böylece her bir ilgi alanına olan odaklanma sayısı, odaklanma süresi verileri elde edilebilmektedir.

3. GÖZ İZLEME TEKNOLOJİSİ

Göz izleme teknolojisi göz bebeği hareketlerinin gelişmiş kamera ve sensörler yardımı ile izlenerek, kullanıcının nereye baktığı, hangi sıklıkla baktığı, ne kadar süre baktığı, nereye odaklandığı gibi nitel ve nicel veriler sunan bir teknolojidir. Bu teknolojinin her ne kadar son yıllarda yaşanan teknolojik gelişmelerle birlikte popüleritesi artmış olsa da araştırma alanı olarak oldukça eskidir. Bu alanda ilk çalışmalar 1879 yılında Louis Emile Javal tarafından yapılmıştır. Javal yaptığı çalışmalarla göz hareketlerinin devam eden ve yumuşak bir geçişle değil, sabitlenmeler (fixations) ve sıçramalarla (saccades) tamamlandığını keşfetmiştir. Bu keşifte tanımlanan sabitlenmeler gözün milisaniyelik duraklamalar yapması, sıçramalar ise gözün iki nokta arasında yaptığı hızlı atlayışlardır. Örneğin, bir cümle okunurken göz ilk kelimedede sabitlenmekte daha sonra ikinci kelimeye sıçrama yapmakta ve bu yeni kelimedede tekrar sabitlenmektedir. Tüm hareket bu şekilde sabitlenme ve sıçramalarla gerçekleşmekte ve okuma işlemi tamamlanmaktadır. Gözün hareketi boyunca yaşanan sabitlenme süresinin artışı, o sabitlenme ile ilgili gerçekleşen beyin aktivitesini ve muhakeme düzeyini artırmaktadır. Görme işlevi de muhakeme düzeyinin artmasıyla gerçekleşmektedir.

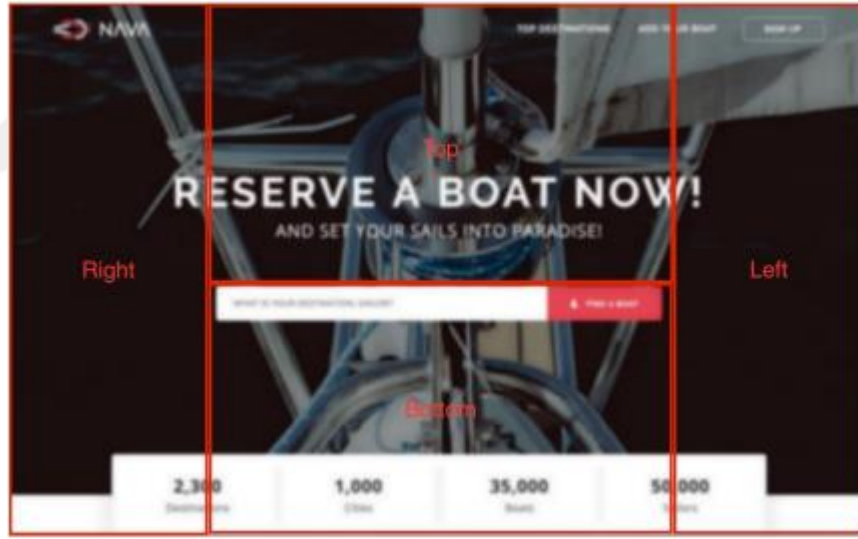
Göz izleme verileri ilk kez 1935 yılında Guy Thomas Buswell tarafından sesli ve sessiz okuma arasındaki farkların incelendiği bir çalışmada kaydedilmiştir [31]. Günümüzde kullanılan göz izleme aygıtlarının ilk versiyonları ise 1990'larda üretilmeye başlanmıştır. 2000'lerde ise bu göz izleme aygıtları üzerinden iş fikirleri üretilmeye başlanmış ve ticari kuruluşlar ortaya çıkmaya başlamıştır.

Günümüzde göz izleme teknolojilerinden pazarlama, sağlık, tasarım, araştırma, eğitim ve daha birçok alanda aktif olarak yararlanılmaktadır. Marketlerde tüketicilerin göz izleme verileri toplanarak, hangi reyonlarda hangi raflara ve hangi markalara bakıldığı incelenmekte, elde edilen sonuçlara uygun pazarlama stratejileri geliştirilmektedir. Kasprowski ve Harezlak tarafından yapılan bir çalışmada [52] otizm ve hiperaktivite tanısı alan çocukların göz izleme verileri toplanarak odaklanma problemleri, yorgunluk belirtileri ve benzeri semptomların analizleri yapılarak erken teşhis ve tedavi yöntemleri araştırılmıştır. Dönmez ve Çağıltay tarafından yapılan bir çalışmada ise [43] görme yetisinde kayıp bulunan çocuklara göz izleme aygıtı ile göz antremanları yaptırılmış, sonuçlardan göz izleme aygıtı kullanımının yararlı olabileceği anlaşılmıştır. Ürün reklamları ile ilgili Özçelik ve Çağıltay tarafından yapılan çalışmada ise katılımcıların göz izleme sonuçlarına göre

reklamların başarı oranları değerlendirilmiştir [16]. Örnek çalışmalardan da anlaşılacağı üzere, göz izleme teknolojisinin geniş bir kullanım alanı bulunmaktadır.

Günümüzde kullanıma sunulan göz izleme aygıtları; kameralar, sensörler ve gelişmiş görüntü işleme algoritmaları ile çalışmaktadır. Sensörler gözlerin kızıl ötesi görüntüsünü oluştururken, kameralar yüksek kare hızında görüntü yakalamaktadır. Elde edilen veriler gelişmiş görüntü işleme algoritmaları tarafından işlenerek detaylar tespit edilmektedir. Bu detaylar üzerinden de gözün konumu ve odak noktası hesaplanmaktadır.

- İlgi Alanı (Area of Interest): Görsel uyarıcı içerisindeki belirli bir alan araştırmacı tarafından çerçeveye alınarak o alan ile ilgili veriler toplanır. Böylece uygulamanın tamamından ziyade daha belirgin özelliklerin analizi yapılabilir. Resim 3.1 'de üst, alt, sağ ve sol olmak üzere ilgi alanlarına ayrılmış bir web sayfası görülmektedir. İlgi alanları bu örnekte olduğu gibi bölgesel olarak belirlenebildiği gibi, öğelerin uygulama üzerindeki alanları gibi daha küçük ve spesifik ilgi alanları da oluşturulabilir.



Şekil 3.17: İlgi Alanlarına Ayrılmış Örnek Bir Web Sayfası [89]

- Odaklanma Süresi (Time Spent): İlgi alanına ne kadar süre ile bakıldığını gösterir. Bu sürenin yüksekliği, bilinçli dikkatin ve ilginin de yüksekliğine işaret eder.

- İlk Odaklanmaya Kadar Geçen Süre (Time to First Fixation): Görsel uyarıcı geldikten sonra ilgi alanına ilk bakılma anına kadar geçen süredir. Bu veri sayesinde dikkat çeken unsurlar belirlenebilir.

- Odaklanma Sayısı (Fixation Counts): İlgi alanına yapılan odaklanma sayısıdır. Bu alan için yapılan hesaplamalarda kullanılabilir.

- Gözlem Oranı (Ratio): Katılımcılardan kaçının ilgi alanına baktığını gösterir.

- Dönen Ziyaretçiler (Revisitors): İlgi alanına ilk odaklanmadan sonra tekrar odaklanma yapan katılımcı oranıdır. Bu veri üzerinden dikkat çekici noktalar tespit edilebilir.

- Sıcaklık Haritası (Heatmap): Görsel ögeye bakma yoğunluğu, odaklanma süresi ve sayısına dair verilerin değerlendirilmesiyle oluşturulan haritadır. Derecelendirme genellikle açık yeşilden kırmızı rene doğru yapılmakta ve kırmızı alanlar en yüksek ilgi alanlarını göstermektedir. Bu harita üzerinden katılımcıların en fazla odaklandıkları alanlar tespit edilebilir. Resim 3.2’de bir resim üzerinde oluşturulmuş sıcaklık haritası görülmektedir. Bu örnek sıcaklık haritasından, katılımcıların çoğunlukla resimdeki mankene odaklanma yaptığı, kırmızı bölgelerden anlaşılmaktadır.



Şekil 3.2: Resim Üzerinde Sıcaklık Haritası Örneği

- Sabitlenmeler ve Sıçramalar (Fixations and Saccades): Göz, hareketi boyunca milisaniye seviyesinde sabitlenmeler yapmaktadır. İki sabitlenme arasındaki hızlı geçişlere de sıçrama denilmektedir. Geniş kullanım alanı nedeniyle çalışmalarda en çok tercih edilen bilgidir. Şekil 3.3’te bir metin üzerindeki sabitlenmeler ve 7 sıçramalar görülmektedir. Bu şekil üzerinde sabitlenmeler daire ile, sıçramalar düz çizgi ile gösterilmiştir.

- Göz Bebeği Büyüklüğü (Pupil Size): Bu veri sayesinde katılımcının odaklanma yapması, görsel uyarıcının heyecan uyandırması gibi analizler yapılabilir.

Eye tracking

From Wikipedia, the free encyclopedia

This article is about the study of eye movement. For the tendency to visually track potential prey, see eye-stalking.

Eye tracking is the process of measuring either the point of gaze (where one is looking) or the motion of an eye relative to eye movement. Eye trackers are used in research on the visual system, in psychology, in psycholinguistics, marketing, and user experience. There are a number of methods for measuring eye movement. The most popular variant uses video images from which the electrooculogram.

Contents	
1	History
2	Tracker types
2.1	Eye-attached tracking
2.2	Optical tracking
2.3	Electric potential measurement
3	Technologies and techniques

Şekil 3.18: Metin Üzerinde Sabitlemeler ve Sıçramalar

- Göz Kapağı Hareketleri (Blinks): Göz kırpma hareketi korkma refleksine, bilişsel iş yüküne veya yorgunluğa işaret edebilir.

Göz izleme aygıtları görüldüğü gibi çeşitli analizlerin yapılabileceği birçok bilgi sağlamaktadır. Ancak her donanımda olduğu gibi göz izleme aygıtlarının da birtakım limitleri bulunmaktadır. Bu limitleri şu şekilde sıralayabiliriz;

- Göz izleme aygıtı katılımcının bakışlarını bir ilgi alanı üzerinde yakalamış olabilir ancak, bu durum ilgi alanına bilinçli bir bakış yapıldığını kesin olarak söyleyemez. Katılımcı farkında olmadan bakışlarını ilgi alanı üzerine getirmiş olabilir.

- Katılımcının herhangi bir görsel öğeyi görmediğini de kesin olarak söyleyemez. Çünkü göz izleme aygıtları çevresel görüş hakkında bilgi verememektedir. Katılımcının ilgi alanına bakışları yakalanmamış ancak çevresel görüşü sayesinde katılımcı bu ilgi alanını görmüş olabilir.

- Her katılımcı için etkin ölçümler gerçekleştirilemez. Örnek olarak, gözlük, lens kullanan veya küçük göz bebeğine sahip katılımcılar için tutarsız sonuçlar üretebilmektedir.

3.1. WEB UYGULAMALARININ KULLANILABİLİRLİK TESTLERİ

İBE'nin artması ve internetin yaygınlaşarak hemen her alanda kullanılmaya başlamasıyla, web uygulamalarının sayıları ve çeşitlilikleri artmıştır. Bu artış ve çeşitlilik uygulamaların hizmet sundukları alanda tercih edilen uygulama olabilmelerini zorlaştırmaktadır. Uygulamanın daha etkin kullanılabilmesi, kullanıcıların uygulamadan memnun olması ve daha fazla kullanıcı

çekebilmek için uygulama üzerinde geliştirilebilecek noktalar devamlı olarak tespit edilmeye çalışılmaktadır. Kullanılabilirlik kavramı tam olarak bu noktada ortaya çıkmaktadır. UPA, kullanılabilirliği “Yazılım, donanım yada herhangi bir ürünün, o ürünü kullanan insanlar için uygunluğunu ve ürünün kolay kullanımını belirleyen ölçüt” olarak tanımlamaktadır. UPA’nın kullanılabilirlik kavramı için yaptığı bu genel tanımın dışında web uygulamaları özelinde yapılmış tanımlar da bulunmaktadır. Jacob Nielsen web uygulamaları için kullanılabilirlik kavramını beş madde ile tanımlamaktadır [17] ;

1. Kolay öğrenilebilirlik
2. Etkin kullanılabilirlik
3. Hatırlanabilirlik
4. Düşük hata oranı
5. Kullanıcı memnuniyeti

Çağltay ise İBE açısından web uygulamalarının kullanılabilirliğini [39] ,bir uygulamada belirlenen görevlerin, hedef kitle olarak belirlenen kullanıcılar tarafından, gerekli eğitim ve teknik desteğin verilmesinin ardından, uygun çevre koşullarında, kolaylıkla ve etkili bir biçimde kullanılabilmesi olarak tanımlamaktadır. Bu tanımlardan anlaşılacağı üzere kullanılabilirlik verimlilik, etkililik ve memnuniyet ifade eden bir ölçüdür.

Ürün ortaya çıktıktan sonra yapılan testler, süreç sonu testler olarak tanımlanmaktadır [39]. Web uygulamalarının kullanılabilirlik testleri de bir süreç sonu testidir. Yaklaşım noktasında ise kullanılabilirlik testleri deneysel yaklaşım sınıfına girmektedir. Deneysel yaklaşım, gerçek bir uygulamayı gerçek görevler ile kullanan kullanıcı verilerini analiz ederek, uygulamayı kullanılabilirlik açısından değerlendirmeyi amaçlar [39]. Bu yaklaşım uygulamaların kullanılabilirliği açısından en gerçekçi sonuçları veren yaklaşımdır. Ancak deneysel yaklaşım, gerçekleştirim noktasında ciddi emek isteyen bir yaklaşımdır.

Kuş ve Çağltay tarafından yapılan bir çalışmada birçok kamu sitesi üzerinde kullanılabilirlik testleri gerçekleştirilmiş, görülen erişilebilirlik problemlerine çözüm önerileri sunulmuştur [49]. Akıncı ve Çağltay tarafından yapılan bir çalışmada ise yine kamu siteleri üzerinde kullanılabilirlik testleri yapılmış, sitelerde görülen problemlere çeşitli tasarım önerileri sunulmuştur [8]. Castilla tarafından yapılan bir çalışmada uygulamanın menü akışında yapılan iyileştirmelerle ileri yaş kullanıcılar için kullanılabilirlik düzeyinin arttığı görülmüştür [9].

Cowen tarafından yapılan başka bir çalışmada ise aynı uygulama için farklı giriş sayfaları üzerinden yapılan ölçümlerle tasarım iyileştirmeleri yapılmış ve uygulamanın kullanılabilirlik düzeyi artırılmıştır [40]. Yapılan örnek çalışmalar ve bu çalışmaların sonuçları, kullanılabilirlik kavramı için yapılan tanımları doğrulayan sonuçlar vermiştir.

Örnek çalışmalarda görüldüğü üzere web uygulamalarının kullanılabilirlik testleri belirli bir süreç içerisinde yapılmaktadır. Bu noktada kullanılabilirlik testleri için genel anlamda şu şekilde bir süreç tanımlamak mümkündür;

1. Kullanılabilirlik testi yapılacak uygulama seçimi yapılır
2. Uygulamaya ve test amacına uygun senaryolar belirlenir
3. Senaryoları gerçekleştirmek üzere uygun katılımcılar bulunur
4. Katılımcılar senaryoları gerçekleştirir
5. Elde edilen sonuçlar değerlendirilerek, gerekli aksiyonlar belirlenir

3.2. WEB UYGULAMALARININ KULLANILABİLİRLİĞİNDE GÖZ İZLEME TEKNOLOJİSİNİN YERİ

Göz izleme teknolojisinin kullanım alanlarından biri de web uygulamalarının kullanılabilirlik testleridir. Yaşanan teknolojik gelişmelerle birlikte göz izleme aygıtlarının kullanım kolaylığı, sağladığı bilgilerin doğruluğu ve tutarlılığı artmıştır. Ayrıca bu aygıtlar sağladıkları çeşitli veri tipleri ile farklı analizler yapma imkânı sunmaktadır. Bu sebeplerden dolayı göz izleme teknolojisinin kullanılabilirlik testlerinde önemli bir yeri vardır ve önemi gün geçtikçe artmaktadır.

Ticari amaçlarla göz izleme teknolojisinden yararlanarak uygulamaların kullanılabilirlik testlerini gerçekleştiren ticari kurumların sayısı da artmaktadır. Uygulama sahipleri uygulamalarının kullanılabilirlik değerlerini öğrenerek, uygulamaları üzerinde geliştirme ve iyileştirmeler yapmak istemektedirler. Bu sayede kullanıcılarının memnuniyetini ve/veya uygulamalarının rekabet koşullarını artırmayı hedeflemektedirler. Bu sebeple, profesyonel olarak bu hizmeti veren kurumlara uygulamalarının kullanılabilirlik testlerini yaptırmaktadırlar.

Ticari amaçların dışında göz izleme teknolojisinden akademik çalışmalarda da yararlanılmaktadır. Web tasarımında kullanıcı tercihlerinin cinsiyete göre belirgin bir farklılık

göstermediği Djamasbi tarafından yapılan bir çalışmanın sonuçlarıyla anlaşılmıştır [62]. Yine Djamasbi tarafından yapılan bir çalışmada [61] Y neslinin (18-31 yaş aralığı) kullanım alışkanları incelenmiş, bu neslin ilgisinin daha çok görsel ve daha az metinden yana olduğu çalışma sonunda görülmüştür.

Goldberg tarafından yapılan bir çalışmada ise göz izleme verileri değerlendirilerek, uygulama için tasarım önerileri hazırlanmıştır [21]. Akademik çalışmalarda da göz izleme teknolojisinden aktif bir şekilde yararlanıldığı incelenen çalışmalardan anlaşılmaktadır.

Göz izleme teknolojisi kullanılarak gerçekleştirilen testler genellikle özelleşmiş laboratuvarlarda yapılır [30], [63], [36]. Bu laboratuvarlarda katılımcılardan kullanılabilirlik testi yapılacak uygulamayı, belirlenen senaryo doğrultusunda kullanmaları istenir. Senaryoyu gerçekleştirmeleri sırasında katılımcıların göz izleme verileri toplanır. Hem ölçüm anında anlık verilerden hem de ölçüm sonrasında kaydedilmiş verilerden analizler yapılır. Bu analizler sonucunda uygulamanın kullanılabilirlik düzeyi belirlenir ve gerekli görüldüğü durumlarda birtakım iyileştirmeler önerilir.

Göz izleme teknolojisi ile elde edilebilecek bilgilerden, web uygulamalarının kullanılabilirliği noktasında yapılabilecek temel analizleri şu şekilde sıralayabiliriz;

- Kullanıcıların uygulamada ilk dikkatini çeken öğeler belirlenebilir.
- Uzun sabitlenmeler incelenerek uygulama üzerinde ilgi gören noktalar belirlenebilir.
- Geri yönlü yapılan sıçramalar incelenerek, kullanıcıda muhtemel kafa karışıklığı yaratmış noktalar tespit edilebilir.
- Sıçrama örüntülerinden kullanıcı davranışlarına yönelik çıkarımlar yapılabilir.
- Resim, metin, video gibi öğelerden en ilgi çekenini tespit edilebilir.
- Uygulama yönlendirmesinin kullanıcı dostu olup olmadığı anlaşılabilir.
- Kullanıcıların uygulama üzerindeki odak bölgeleri belirlenebilir.

3.3. KULLANILABİLİRLİK TESTLERİ İÇİN GÖZ İZLEME METRİKLERİ

Kullanılabilirlik testlerinin göz izleme teknolojisinden önce de var olduğu bilinmektedir. Dolayısıyla ölçüm ve değerlendirme noktasında kullanılan, göz izleme verilerinden

yararlanılmayan metrikler bulunmaktadır. Bazı çalışma veya arařtırmalarda halen kullanılabilen bu metrikler, daha temel metriklerdir ve sađladıkları bilgi kısıtlıdır.

Göz izleme teknolojisi kullanılması durumunda, metrikler üzerinden daha iyi deđerlendirmeler yapmak mümkündür. Bir örnek üzerinden düşünülecek olursa, kullanıcılardan bir uygulama üzerinde senaryo gerçekleştirilmesi istenir ve senaryo gerçekleştirimi için harcanan süre ölçülür. Harcanan süreler üzerinden çeşitli analizler yapılır. Makul gibi gözükse de bu metriğin eksik bir yönü bulunmaktadır. Metrik bu haliyle sadece zamanı ifade edebilmektedir. Ancak aynı metrik göz izleme verileriyle deđerlendirilecek olursa çok daha anlamlı sonuçlar elde etmek mümkün olacaktır. Örneğin bu metrik göz izleme verileriyle deđerlendirilerek, kullanıcının dikkatinin tüm süre boyunca uygulamada olup olmadığı anlaşılabilir.

Göz izleme verileri üzerinden, web uygulamalarının kullanılabilirlik problemlerinin tespit edildiđi bir çalışmada bu alandaki çalışmalara yönelik literatür taraması yapılmıştır. Literatür taraması sonucunda, kullanılmakta olan göz izleme metrikleri incelenmiştir [4]. Bu metriklerin kullanılabilirlik çalışmalarında ne amaçla hangi problemin çözümüne yönelik kullanılacağı, verinin tipine göre kategorilendirilmiştir. Erdemir'in kitabında belirttiđi bazı metriklerin de bu kategorilendirme arasında yer aldığı görülecektir [37].

Sabitlenme (fixation) ile ilişkili metrikler řu şekildedir

- Hedefe ilk sabitlenmeye kadar geçen süre: Hedefin dikkat çektiđini göstermesi noktasında bu sürenin kısa olması kullanılabilirlik açısından iyidir.
- Konumsal sabitlenme dağılımı: Uygulama üzerindeki sabitlenmelerin dağılımının deđerlendirilmesidir. Kullanılabilirliđi yüksek bir uygulama için bu dağılımın az olması beklenir. Konum üzerine düşen sabitlenme sayısı ile ifade edilebilir.
- Sabitlenme süresi: Sabitlenmenin başlangıç ve bitiři arasındaki süre ile ifade edilebilir. Deđerin uzun olması dikkat çekici veya anlaşılması güç bir görsele işarettir. Farklı metriklerle birarada deđerlendirilmesi daha dođru sonuçlar verecektir.
- Toplam sabitlenme sayısı: Senaryo üzerinden ölçüm yapıldığında, yüksek olması senaryo gerçekleştiriminin zor olduđunu gösterir ve kullanılabilirlik düşüklüğüne işarettir.
- Tekrarlanan sabitlenmeler: Görsel içeriğin bozukluđunu veya anlamsızlıđını gösterir. Kullanılabilirlik düşüklüğüne sebep görülebilecek bir durumdur.

- İlgi alanı için kullanıcıların sabitlenme oranı: Bir ilgi alanında sabitlenme oranının yüksek olması dikkat çekici görsel içeriğe işaret eder. İlgi alanı üzerine düşen sabitlenme sayısı ile ifade edilebilir.

- Sıçrama-sabitlenme oranı: Uygulama üzerinde daha çok işlem yapıldığını, taramanın daha az olduğunu gösterir. Sıçrama sayısının sabitlenme sayısına oranlanması şeklinde ifade edilebilir.

- Sabitlenmeler arası uyum: Yakın veya aynı ilgi alanlarında sabitlenmelerin görülmesi tasarım özelliklerinin başarısını gösterir.

- İlgi alanındaki sabitlenme süresi: İlgi alanındaki sabitlenmenin başlangıç ve bitiş süreleri arasındaki farkla ifade edilebilir. Yüksek oluşu, görseli yorumlamada zorluk olduğunu gösterebilir.

- İlgi alanındaki sabitlenme yüzdesi: Önemli veya dikkat çekici bir görsel elemana işaret eder. İlgi alanındaki toplam sabitlenme sayısının toplam sabitlenme sayısına oranlanması ile ifade edilebilir.

Sıçrama (saccade) ile ilişkili metrikler şu şekildedir;

- Sıçrama sayısı: Sıçrama sayısının artışı kullanıcının aradığı hedefi bulamadığını ve uygulamada tarama yaptığını gösterebilir.

- Sıçrama yönünde görülen değişiklik: Kullanıcının hedef değiştirdiğini veya tasarımın kullanıcı beklentisine uymadığını gösterir.

- Sıçrama büyüklüğü: Anlamlı, dikkat çeken görsel öğeye işaret eder. Takip eden iki sıçrama arasındaki uzaklıkla ifade edilebilir.

- Sıçrama süresi: İki sıçrama arasındaki süre ile ifade edilebilir. Düşük olması görsel kalitesinin düşüklüğünü gösterebilir.

Birçok metrikte görüldüğü üzere, sonuçlar üzerinden kesin değerlendirmeler yapmak mümkün olamamaktadır. Bu nedenle çalışmalarda birden fazla metrik bir arada değerlendirilmektedir.

3.4. GÖZ İZLEME TEKNOLOJİSİ VE KULLANIM ALANLARI

Göz hareketlerinin izlenmesi üzerine yapılan arařtırmalarda temel olarak “eđer kullanıcınn nereye baktıđını bilirsek, o anda tam olarak ne düşündüğünü bilemesek bile en azından ne hakkında düşündüğünü anlamamıza yardımcı olabilir” düşüncesi kullanılmakta ve bu düşünce arařtırmacıların birçok konuda uygulama geliřtirebilmesine olanak sağlamaktadır. Göz izleme teknolojisinin muhtemel uygulama alanları ve örnekleri ařađıda sıralanmaktadır:

3.4.1. İnsan Bilgisayar Etkileřimi

İnsan bilgisayar etkileřiminde göz izleme tekniđinden özellikle kullanılabilirlik çalışmalarında faydalanılmaktadır. Göz izleme sayesinde geliřtirilen uygulamalara ait arayüzlerin tasarım ařamasında, tasarım süreci ierisinde veya sonrasında kullanıcının tasarıma yaklařımı, nasıl kullandığı ve ne tür sorunlarla karřılařtığı hakkında bilgi sahibi olunarak tasarım geliřtirilebilmekte ve deđiřtirilebilmektedir. Göz izleme cihazları ile kullanıcıdan komut alınabilmekte, fare ve tuř takımı ile yapılan işlemler göz hareketleri ile sağlanabilmektedir. Kullanıcının yoğunlařtığı alanda bulunan uygulamalar arasında geişler sağlanabilmekte ve yazı büyüklükleri otomatik olarak büyütülerek kullanıcıya kolaylıklar [77].

3.4.2. Pazarlama ve Reklâm

Bir müşterinin dikkatini en çabuk çeken ürünler hangileridir sorusu pazar arařtırmacılarının en çok cevap aradığı soruların başında gelmektedir. Örneđin bir süpermarkette müşterilerin ilk olarak baktıkları reyonların tespit edilebilmesi, hangi řekil ve renkteki ürünlerin daha çok dikkat çektiđinin bulunması deđiřik pazarlama stratejileri geliřtirmek aısından oldukça önem taşımaktadır. řekil 3.4’de böyle bir arařtırmada göz izleme cihazı ile reyonda bulunan ürünlerin dikkat çekme analizlerinde ortaya çıkan sonuç görölmektedir. Bu sonuçlara göre göz hareketlerinin, göz hizasında ve reyonun ortasında bulunan ürünlerde daha fazla yoğunlařtığı görölmektedir.



Şekil 3.4: Reyondaki Ürünlere Bakış Sırası

Benzer bir şekilde bir sayfa düzeni editörü için kullanıcının en çok dikkatini çeken resim, yazı formatları, başlıklar oldukça önemlidir. Veya bir web sayfası tasarımcısı için alınan reklâmların yeri, bu reklâmların gösterim tarzı, bir ziyaretçinin en çok neyi okuduğu, belirli bir sayfada ne kadar kaldığı, daha sonra hangi bağlantıya tıkladığı gibi sorular da oldukça önem arz etmektedir. Göz izleme tekniği sayesinde yukarıda bahsedilen soruların çoğuna cevap alabilmek mümkün gözükmemektedir. Şekil 3.5’de bir web sayfasının göz izleme tekniği ile yapılan analizine ait sonuç gösterilmektedir. Bu sonuca göre bilgilerin arasında belirli bir mesafenin korunmasının görsel algıyı kolaylaştırdığı gözlemlenmiştir. Kırmızı renk en uzun bakılan yerleri, sarı ve yeşil sırasıyla kısa ve daha kısa bakılan yerleri göstermektedir.



Şekil 3.5: Resimlere Ait Odaklanma Oranları

3.4.3. Ergonomi ve İşyeri Tasarımı (Human Factors)

Güç istasyonları ve trafik kontrol sistemleri izleyici personellerinin, uzun yol motorlu taşıt sürücülerinin; konsantrasyon eksikliği ve uyuklama hallerinde göz izleme tekniğinden bir nevi erken uyarı sistemi veya güvenlik cihazı olarak faydalanılmaktadır. Ayrıca benzeri görevleri icra eden izleyici operatörler için göstergelerin ve kontrol odalarının tasarımlarında göz izleme tekniği ile yapılan analizlere başvurulabilmektedir [18], [67].

3.4.4. Özürlülere Yardımcı Olan Uygulamalar

Özürlü insanların yaşamları onların çevreleri ile iletişim kurmalarını sağlayarak çok daha iyi hale getirilebilmektedir. Konuşamayan ve ellerini kullanamayan, ellerini kullanamadığı için yazı dahi yazamayan engelliler gözlerini kullanarak, göz izleyicisi sistemlerin bağlı olduğu bilgisayarlar aracılığıyla çevreleri ile iletişim kurabilmektedirler. Dahası oyun oynayıp, onları mutlu edebilecek çeşitli uygulamaları da çalıştırabilmektedirler. Şekil 3.6'da göz izleme cihazını kullanarak okulunda arkadaşları ile konuşabilen, ev ödevlerini yapabilen ve hatta email gönderebilen engelli bir çocuk görülmektedir.



Şekil 3.6: Göz İzleme Cihazı ile Çevresiyle Etkileşen Engelli Bir Çocuk ve Babası

Bu sistemde engelli kişinin tekerlekli sandalyesine monte edilen sistem sayesinde göz hareketleri ile önündeki monitör üzerinden işlemler gerçekleştirmesine olanak sağlanmaktadır. Konuşamayan ve ellerini hareket ettiremeyen bu kişinin belirlenen sınır değerini aşan bir süre boyunca aynı harfe bakarak, o harfi seçmesi sağlanır. Yazmak istediği ifadeyi bitirdiği vakit, gözünü ekrandaki KONUŞ butonu üzerinde, sınır değeri aşacak bir süre boyunca tutarak, yazdığı metnin seslendirilmesini sağlayabilmekte ve böylece çevresi ile etkileşim kurabilmektedir.

3.4.5. Psikoloji, Psikofizik Ve Nöroloji

Nöroloji ve psikoloji ile ilgili arařtırmalarında, insan görme sistemi ve bilişsel süreçlerin anlaşılması konularında göz izleme tekniğinden faydalanılmaktadır [71]. Nörolojik bozuklukların anlaşılmasında bu teknikten faydalanılarak çeşitli bulgular elde edilmiştir.

3.4.6. Grafik, Sanal Gerçeklik Ve Simülatör Uygulamaları

Üretim hattı bozukluklarını tespit etme, X-ray ve MR cihazlarına ait görüntüleri yorumlama gibi tecrübe ve hassasiyet gerektiren görevlere yeni başlayacak olan personelin eğitiminde, tecrübe sahibi operatörlerin bu görevleri gerçekleştirirken sergiledikleri görsel tarama davranışlarının ortaya çıkarılmasında göz izleme tekniğinden faydalanılmaktadır [7].

Benzer şekilde tecrübeli uçuş pilotlarının kritik göstergelere (MFD, HUD ve radar ekranları gibi) hangi sıklıkla ve ne sıra ile baktıkları gibi bilgiler de göz izleme teknikleri ile elde edilebilmektedir. Çok yüksek hızlarda milisaniyelerle ölçülen zaman aralıklarında verilen bu tepkiler oldukça önemli olabilmektedir [56]. İşte bu nedenle benzer görevlerde yer alacak yeni personellerin eğitiminde göz izleme tekniğinden faydalanılmaktadır.

Şekil 3.7’te acil iniş, kalkış hareketleri gibi yoğun tecrübe gerektiren durumlarda, eğitimi gerçekleştirilen kişinin dikkatini nereye yoğunlaştırdığının izlenebildiği, gerektiği takdirde hatasını tespit ederek, henüz eğitim aşamasında iken gereken müdahalenin yapılabildiği bir sistem görülmektedir [12]. Bu sistemde eğitimdeki personelin kafasına monte edilen bir göz izleme cihazı ile tüm göz hareketleri toplanmakta ve değerlendirilebilmektedir.



Şekil 3.7: Uçuş Eğitim Simülatoründe Bulunan Bir Göz İzleme Sistemi

Benzer sistemler sürücü dikkati gerektiren çeşitli uygulamalarda da kullanılmaktadır. Şekil 3.8’de gösterilen sistemde bir sürücünün göz hareketleri bir kamera vasıtası ile izlenmekte, dikkati gözlenmekte, sahne üzerinde hangi noktaya baktığı dışarıdan gerçek zamanlı olarak izlenebilmektedir.

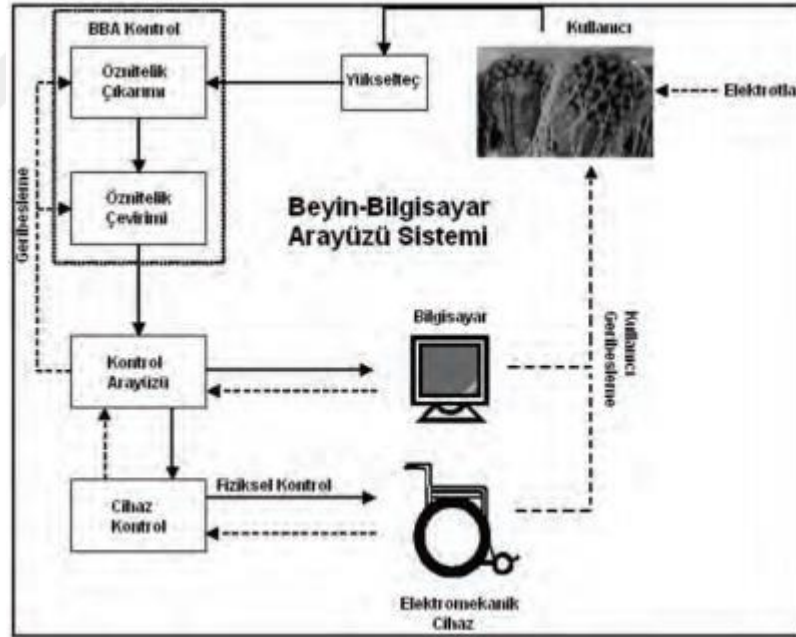


Şekil 3.8: Sürücü Eğitiminde Kursiyerin Göz Hareketleri Tespiti

Önceki kısımlarda bazı uygulama örnekleri verilen göz izleme tekniği daha birçok alanda kullanılmaktadır ve 20. yüzyılın başlarında temelleri atılan bu teknik, gelişen teknolojinin de yardımı ile daha birçok alanda kullanılacaktır [71], [50].

4. BEYİN BİLGİSAYARI ARAYÜZÜ

Beyin Bilgisayar Arayüzleri (BBA), insanların motor sinir sistemlerini kullanmaksızın bir bilgisayarı, elektromekanik bir kolu ya da çeşitli nöroprotezleri kullanmalarını olanaklı hale getiren sistemlerdir. Beynimizi oluşturan çok sayıdaki sinir hücresi (nöronlar) elektrokimyasal etkileşimler ile birbirleriyle haberleşirler. Bu haberleşme sırasında, kafa derisi üzerine yerleştirilecek elektrotlar sayesinde, meydana gelen iletişim hakkında bilgi edinmek mümkündür. Beynin, basitçe, farklı işlemleri gerçekleştiren bölümlerden oluştuğunu düşündüğümüzde ilgili bölüme yakın yere yerleştirilecek olan elektrotlar o bölgeye ilişkin bir bilgi edinmemizi sağlarlar. BBA sistemleri farklı kombinasyondaki elektrotlar ve bu elektrotlardan alınan elektriksel sinyallerin farklı şekillerde analiz edilmeleri ile gerçekleştirilirler. Bu sistemler motor fonksiyonlarını yerine getiremeyen Amiyotrofik Lateral Skleroz (ALS) veya Tetrapleji hastaları gibi kişilerde rehabilitasyon amaçlı kullanırken, diğer insanlar için bilgisayarlarını yada herhangi bir elektronik kontrol sistemini kullanmalarını sağlayabilir.



Şekil 4.1 : Genel bir Beyin Bilgisayar Arayüzü Şeması

BBA sistemlerinde elektrotlar ile nöronların elektrokimyasal etkileşimlerinin elektriksel yansımalarını algılayabilmenin 3 yolu vardır. Birinci yöntemde elektrotlar mikroelektrotlar halinde beyin kabuğu üzerine direkt olarak yerleştirilirler. Bu yöntemde kafatasının açılması ve bir operasyon ile mikroelektrot matrisinin beyine tutturulması sağlanır. İkinci yöntemde kafa derisi yine açılır fakat, bu sefer elektrotlar beyin kabuğuna saplanarak yerleştirilmez,

yine elektrot matrisi halinde beyin kabuğu üzerine serilir. Buradaki elektrotlar ilk yöntemdeki gibi mikro yapıda değildir. Üçüncü ve son yöntemde ise kafatası üzerine iletken bir jel ile tutturulan elektrotlar kullanılır. Bu yöntemler içerisinde en pratik olan üçüncü yöntemdir. Şu anki araştırmaların çoğu bu yöntemi kullanan BBA sistemleri üzerinedir.

Elektrotlardan gelen düşük genlikteki elektrofizyolojik sinyal ilk olarak bir yükselteç bloğuna gelir. Burada yükseltile sinyal belli bir ilgili frekans aralığında süzülür. Süzülen sinyal üzerinde, hangi sinyallerin ne tür anlamlar içerdiklerini anlamak üzere çeşitli öznitelik çıkarımları uygulanarak, ne tip örüntülerin ne tip anlamlar içerebilecekleri araştırılır. Gerekli özniteliklerin bulunmasının ve bu özniteliklerin anlamlandırılmasının ardından gerekli kontrol komutları üretilir ve yapılması arzu edilen motor fonksiyon gerçekleştirilir.

BBA sistemleri halen geliştirilme aşamasındaki sistemler olup ülkemizdeki üniversiteler de dahil olmak üzere dünyada 100'ün üzerinde okul ve araştırma kurumu tarafından çeşitli seviyelerde incelenen oldukça popüler bir araştırma alanıdır. Bu araştırmalar sonucunda dünya nüfusunda önemli derecede yüksek sayıdaki felç hastaları için hayatlarını daha kaliteli yaşayabilme olanağı sunulabilecektir.

Teknik Olarak ;

Günümüzdeki BBA sistemleri derin ve yüzeysel EEG (Elektroensefalografi) kayıtları ile alınan beyin sinyallerinin işlenmesi ile gerçekleştirilmektedir. Yüzeysel kayıtlar daha kolay ve pratik yapılabildiklerinden dolayı en çok tercih edilen BBA yöntemi haline gelmişlerdir. Gümüş-Gümüş Klorür (Ag-AgCl) elektrotlar ile kafatasından alınan sinyaller önce bir biyo potansiyel yükselteci ile yükseltilir, ardından filtrelenerek ilgi boyutu daraltılır. Daha sonra kişilere gerçekleştirilmesi istenilen beyin aktivitesine göre çeşitli görevler verilir ve bu görevler esnasında yapılan EEG kayıtlar makine öğrenmesi algoritmaları geliştirmek için kullanılırlar. Destek Vektör Makineleri, Yapay Sinir Ağları ya da Gizli Markov Modelleri gibi farklı yöntemler kullanılarak geliştirilen sınıflandırıcılar, kişilerin sistem ile gerçekleştirdikleri birkaç deneme ile eniyileştirilirler (Optimization). EEG tabanlı BBA sistemler dışında, tek hücre kayıtlamaları (SCR), bölgesel alan potansiyelleri (LFP), elektrokortigografi (ECoG), magnetoensefalografi (MEG), fonksiyonel manyetik rezonans görüntüleme (fMRI), yakın infrared spektroskopisi (NIRS) gibi teknikler de BBA geliştirmek için kullanılan görüntüleme teknikleridir. Fakat bu tekniklerin hiç biri EEG kadar çabuk erişilebilir, ucuz ve pratik

değildirler. Bundan dolayıdır ki gelecekteki BBA sistemlerinin temelinde de EEG tabanlı sistemleri sık sık görmemiz olasıdır.

Şimdiye kadar gerçekleştirilen en başarılı örnekler Brown Üniversitesi'nde Donogue ve arkadaşlarının beyne invazif olarak yerleştirdikleri elektrotlar ile gerçekleştirdikleri BBA sistemleri ve Wadsworth Center, Albany de Mc. Farland ve arkadaşlarının felçli hastalar için gerçekleştirdikleri EEG tabanlı BBA sistemleridir. Bu sistemlerden bir tanesinde eskiden bilimadamı olan fakat bir kaza sonucu alt ve üst uzantılarını (ekstremiteler) kullanamaz durumdaki bir kişinin bilgisayarını kontrol edebilir hale gelmesi sağlanmıştır. Bu kişi geliştirilen sistem yardımıyla makaleleri kendi açıp okuyabilmekte, bilgisayarda çeşitli uygulamaları yapabilmekte ve en önemlisi kendi makalelerini bir çeşit P300 yazı yazma sistemi kullanarak yazabilmektedir. P300'ün anlamı beyne verilen bir etkinin beyinde yaklaşık 300ms sonra bir tepki oluşturması üzerine verilen etkiye paralel olarak beklenen sinirsel aktivite değişimidir. Bu yöntemin uygulaması olarak ilk defa Donchin ve arkadaşları tarafından gerçekleştirilen P300 yazı yazma sisteminin çalışma prensibi şu şekildedir. Bilgisayar ekranda klavyede bulunan tüm harfler ve bazı özel işaretler bir matris halinde belirir. Bu matrisin satır ve sütunları etrafında belirli zaman aralıklarıyla rastgele gezinen bir dikdörtgen kutucuk beyinde ilgili kutunun içerisindeki harflerin yazmak istediğimiz harf olup olmasına göre bir tepki sinyali oluşturur. İlgili tepkiler algoritma tarafından değerlendirilir ve kişinin yazı yazması sağlanır. Çok hızlı bir şekilde yazı yazılmasını mümkün kılmasa da o durumdaki bir insan için önemli bir iletişim kaynağı durumundadır.

BBA sistemleri Avrupa Birliği tarafından da son zamanlarda PRESENCIA ve MAIA gibi projelerle de desteklenen önümüzdeki yüzyıl içerisinde araştırmaya açık önemli bir bilimsel konu durumundadır.

Görüldüğü üzere EEG kayıtları ile BBA tasarımı partik olmakla birlikte uzun süreli kullanılabilir değildir. Uygulama için kafaderisi her seferinde elektrotlar düzgün şekilde yerleştirilmelidir. Bu işlem kullanıcı için rahatsız bir boyuta ulaşabilir.

Buna karşın herhangi bir elektrot yerleşimi gerekmeden göz hareketleri ile benzer beyin-makine etkileşimi gerçekleştirilebilir. Emotiv Epos cihazı ile beyin dalgası frekans bilgilerinin hızlı Fourier dönüşümü (FFT) yöntemi ile analizi yapılmış ve kişiyi psikolojik durumu tespit edilip dikkat seviyesinin ölçülmesi sağlanmıştır [86].

5. DENEYSEL ÇALIŞMAYA GİRİŞ

Göz izleme sisteminin temel kullanımını, kullanıcının gözlerinin sabit hareketini, durmadan ve belirli bir alana odaklanmadan önce bir ekrana bakarken kaydetmek ve analiz etmektir. $20 \times 1,9 \times 1,9$ cm ölçülerinde, dünyanın en küçük göz izleme cihazı olarak kaydedilen eye-tribe cihazı bu çalışmada test amaçlı kullanılmıştır. Bu göz izleyicinin ayrı bir güç kaynağı gerektirmediğinden, daha da taşınabilir olmasını sağlanmıştır.

Eye Tribe, USB 3.0 Super Speed bağlantı noktasına ihtiyaç duymaktadır; bu da onu yüksek hızlı bilgisayarlarla ve akıllı telefonlarla, tabletler gibi cihazlarla uyumlu hale getirmektedir. Eye Tribe, C ++, C # ve Java programlama platformları için mevcut basit bir yazılım geliştirme seti ile birlikte Microsoft Windows 7 veya daha yeni ve OSX ile uyumludur.

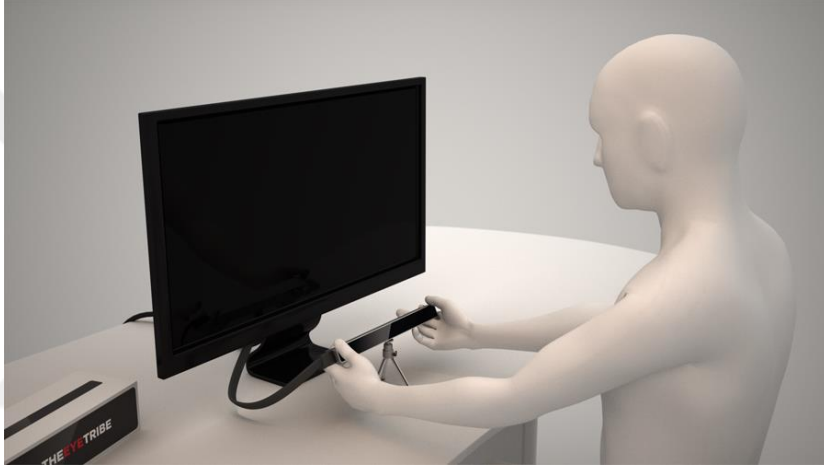
Eye Tribe izleyicinin ana bileşenleri, bir akıllı telefon, bilgisayar vb. içinde kolayca yerleştirilebilen bir kamera ve yüksek çözünürlüklü kızılötesi LED'den oluşur. Cihaz, göz hareketini izlemek için bir kamera kullanır. Kamera, görüntüleri alarak ve bilgisayar-görüntü algoritmalarına tabi tutarak, kullanıcıların göz bebeklerinin en küçük hareketlerini bile izler. Cihaz içeride en iyi şekilde çalışmasına rağmen, aynı zamanda donanım, kamera sensörleri ve çalışma ortamındaki farklı ışık ayarları ile çalışırken tatmin edici sonuçlar verebilmektedir.

5.1. CİHAZ SABİTLENMESİ

Eye Tribe cihazda sabitleme belli bir noktada gözlere odaklanmaktadır. Bu gözlere odaklanma ortalama 100-1000 ms arasında olan ve çoğunlukla 200-500 ms arasında olan ve işlenen bilginin kalitesi ile mevcut bilişsel yüke bağlı olan düzeltmeler yapmaktadır. Düzeltmeler, gözlerin göreceli olarak durağan görüldüğü, bilgiyi girdiği veya “kodladığı” durumlardır. Senkron için gerekli olan cihazın kurulması ve konumlandırılması aşağıda Şekil 5.1, Şekil 5.2 ve Şekil 5.3’de anlatılmaktadır.



Şekil 5.1 : Eye Tribe İzleyicinin 3.0 ile USB Bağlantısı [80].



Şekil 5.2 : Pritop ile Bilgisayar – Eye Tribe İzleyicinin Mesafe Ayarlanması [80].



Şekil 5.3 : Eye Tribe İzleyici ile Denekler Arasında Oturma Konumunun Ayarlanması [80].

Eye Tribe izleyici ile bilgisayar arasındaki tüm bağlantı işlemleri yapıldıktan sonra cihaz açılır. Sistemde herhangi bir sorun olmaması durumunda cihaz bize başarılı bir şekilde açılarak aktif

edildiğinin bilgisini vermektedir. Şekil 5.4’de hazır konumda olan sistemin çıkış bilgisi yer almaktadır.

```
C:\Program Files (x86)\EyeTribe\Server\EyeTribe.exe
[13:07:45.755] INF: Initializing environment
[13:07:45.755] INF: Constructing logfile:
C:\Users\leon\AppData\Local\EyeTribe\Logs\20190205130745_54346357.log
*****
** The EYE TRIBE TRACKER Server **
*****
Version: 0.9.77

Settings applied:
TCP port: 6555
Connections: local only
Framerate: 30

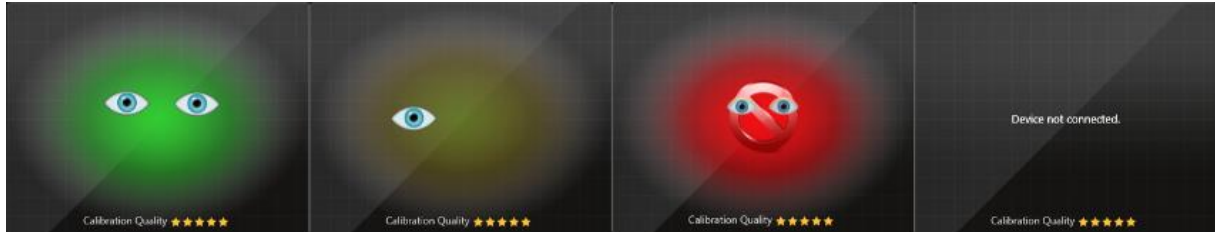
Initializing the Eye Tribe Server
[13:07:46.856] INF: Default calibration profile loaded!

The Eye Tribe Tracker is waiting to be connected!
The Eye Tribe Tracker firmware revision is 293
Accessing Tracker device

The Eye Tribe Tracker stands ready!
```

Şekil 5.4 : Eye Tribe İzleyici Config Ayar Yapılandırması Sonuç Çıktısı

Kullanmakta olduğumuz cihaz öncelikli olarak her işlemde sekron edilmek zorundadır. Yapılan her başarılı kalibrasyon işlemlerinden sonra cihazımız bize senkron kalitesi hakkında bilgi veren bir resimli mesaj çıkarmaktadır. Şekil 5.5’te bu gösterilmektedir.



Şekil 5.5 : Senkron İşlemlerinde Başarı Durum Gösterimi

5.2. EYE TRİBE İZLEYİCİ İLK VERİ AKIŞI VE EKRANDA GÖSTERİLMESİ

Eye Tribe izleyici çalıştırıldığında belirli bir algoritma ile ekran çıktısı üretmektedir. Bu ekran çıktısını elde edebilmek için cihazı üreten firmanın geliştirmiş olduğu SDK(Software Development Kit) sisteminden faydalanmaktayız.

SDK (Software Development Kit), donanım ve yazılım geliştiricileri için sunulan ve içerisinde bir takım yazılım geliştirme araçlarını barındıran geliştirme aracıdır. SDK’lar sıklıkla API, örnek kodlar ve dokümantasyon verilerini içerisinde barındırmaktadır. Firmanın kurumsal SDK

(Software Development Kit) kütüphanesini projelerimize eklenmelidir. Kullanılan editörlerde bu yükleme işlemleri farklı adımlar kullanılarak gerçekleştirilmektedir.

Sistemde ilk veri akışının ekran çıktısı almak ve yapıyı anlatmak için Şekil 5.6'te akış diyagramı kullanılmaktadır. Burada veriler SDK (Software Development Kit) içerisinde yer alan TCP (Transmission Control Protocol) socket bağlantı fonksiyonu çağrılarak cihaza iletişim gerçekleştirilir. Yapılan bu bağlantı başarılı bir şekilde gerçekleşirse cihaz tarafından veri akışı yayınlanmaya başlatılır. Eye Tribe izleyici cihazından alınan verileri çeşitli kayıt ortamlarında kaydederek bilginin saklı kalması sağlanmaktadır. Veriler 10 milisanyeden daha az bir periyot içerisinde güncellenmektedir. Kod kısmında sistemin veri akışını sağlayan socket bağlantısını kapatmadığımız sürece bilgilerin gelişi devam edecektir.



Şekil 5.6 : Eye Tribe İzleyici Veri Akış Algoritması

Uygulama geliştirme ortamım windows ve yazılım dili olarak C# kullanılmıştır. Geliştirmiş olduğum bu arayüzde çok yüksek hızda veri akışları elde edilmiştir. Eye Tribe izleyici çalıştırılarak Şekil 5.7'de elde etmiş olan çıktı yer almaktadır.

Form1

KAYIT SAYISI 925

```
{ "category": "tracker", "request": "get", "statusCode": 200, "values": { "frame": { "avg": { "x": 389.4839, "y": 250.3150, "fix": false, "lefteye": { "avg": { "x": 304.0268, "y": 235.0094, "pcenter": { "x": 0.3261, "y": 0.5665, "psize": 17.6344, "raw": { "x": 304.0268, "y": 235.0094 } }, "righteye": { "avg": { "x": 389.4839, "y": 250.3150, "pcenter": { "x": 0.3261, "y": 0.5665, "psize": 17.6344, "raw": { "x": 304.0268, "y": 235.0094 } } } } } } } }
```

Category	Request Sorgu	statusCode	ValuesParametre	X	Y	Fix	Lefteye	LefteyeY
tracker	get	200	frame.avg.x:404....	0	0	<input checked="" type="checkbox"/>	avg.x:396.0819	651.8685
tracker	get	200	frame.avg.x:403....	0	0	<input checked="" type="checkbox"/>	avg.x:395.0807	652.9588
tracker	get	200	frame.avg.x:403....	0	0	<input checked="" type="checkbox"/>	avg.x:394.3188	653.8548
tracker	get	200	frame.avg.x:402....	0	0	<input checked="" type="checkbox"/>	avg.x:392.6974	654.8354
tracker	get	200	frame.avg.x:402....	0	0	<input checked="" type="checkbox"/>	avg.x:391.2093	655.4937
tracker	get	200	frame.avg.x:0.0	0	0	<input type="checkbox"/>	avg.x:0.0	0.0
tracker	get	200	frame.avg.x:0.0	0	0	<input type="checkbox"/>	avg.x:0.0	0.0
tracker	get	200	frame.avg.x:401....	0	0	<input checked="" type="checkbox"/>	avg.x:0.0	0.0
tracker	get	200	frame.avg.x:765....	0	0	<input type="checkbox"/>	avg.x:0.0	0.0
tracker	get	200	frame.avg.x:608....	0	0	<input type="checkbox"/>	avg.x:0.0	0.0
tracker	get	200	frame.avg.x:576....	0	0	<input type="checkbox"/>	avg.x:388.2586	342.8936
tracker	get	200	frame.avg.x:536....	0	0	<input type="checkbox"/>	avg.x:404.3786	343.3409
tracker	get	200	frame.avg.x:516....	0	0	<input type="checkbox"/>	avg.x:400.1070	344.1526
tracker	get	200	frame.avg.x:492....	0	0	<input type="checkbox"/>	avg.x:397.9375	342.3191

Göster Bitir

Şekil 5.7 : Eye Tribe İzleyici İlk Ekran Çıktısı

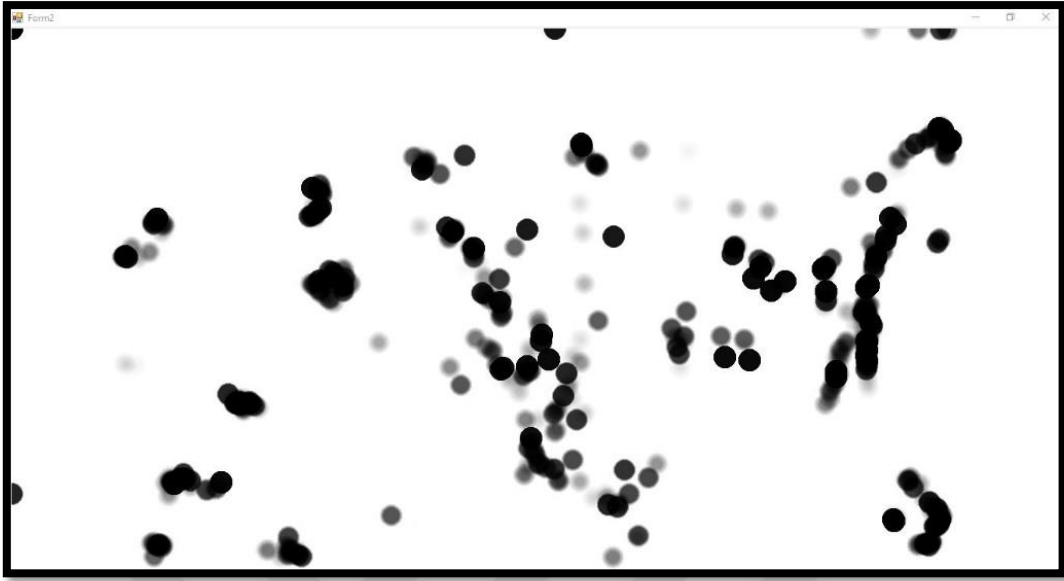
Göster Buton tıkladığımızda Connect metodu ile cihaza bağlantı sağlanır. Cihaz bağlantısı başarılı olmadığı durumda sistem mili saniye bazında tekrar başarılı bir bağlantı olana kadar denemeye devam eder. Bağlantı işlemi başarılı olduktan sonra ikinci aşamaya geçilir. Bu aşamada Send metod çağrılır. Bu metod içerisinde get keywords gönderilerek cihaz içindeki verileri sürekli olarak almamıza olanak sağlayan sistem aktif edilmektedir. Çalışan sistem kapatılmadığı sürece veri akışımız devam edecektir. Bir sonraki aşamada yer alan ListenerLoop parametresiz ve geri dönüşsüz olan metod çağrılır. Çağırduğumuz bu metod Eye Tribe izleme cihazının bize sağlamış olduğu verileri json formatında devamlı almamıza olanak sağlamaktadır.

ListenerLoop metodundan sonra form üstüne konulmuş olan timer zamanlayıcı ile GetListAsync() metodu çağrılır. Bu metod ile gelen verileri txt formatında metin olarak kayıt edilir. Bu saklama yordamı yapılırken cihaz tarafından bize gönderilen json türünde veri JsonStringToDataTable parametre olarak string türünde veri alan fonksiyon ile veri ayrıştırma işlemine tabi tutulur. Veri ayrıştırma işlemi tamamlandıktan sonra data, ekran çıktısı olarak projede gösterilmektedir. Proje içine konulmuş olan label etiket türüne toplam okunan kayıt sayısı yazılır. Bitir buton tıkladığımızda sistem tüm işleri kapatarak görevini tamamlamaktadır.

5.3 EYE TRİBE İZLEYİCİ İLE SICAKLIK HARİTASI ÇIKARTMAK

Eye Tribe izleyici ile bilgisayar ekranında kullanıcının hangi noktalara odaklandığının tesbitini sağlamak için sıcaklık haritası çıkarılmıştır. Şekil 5.8'de bu yapılan çalışmanın görsel bölümü

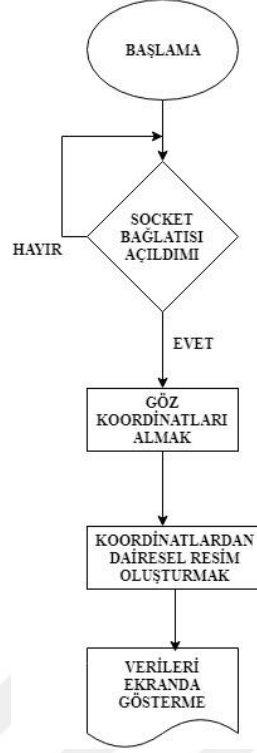
yer almaktadır. Bu çalışmada kullanıcı ekranda en fazla nerelere odaklanırsa o alanlar daire şeklinde gösterilmiştir.



Şekil 5.8 : Sıcaklık Haritası

Yukarıda yer alan şekilde koordinat düzleminde x, x_1 ve y, y_1 eksenler baz alınmıştır. Dairesel alanların çiziminde radius olarak x ve y koordinat sistemi baz alınmıştır.

Eye Tribe izleyici ile sıcaklık haritası elde etmek için kullandığımız kod bloğunun algoritmik çıktısı Şekil 5.9'de yer almaktadır.



Şekil 5.9 : Sıcaklık Haritası Akış Şeması

Çalışma sistemimiz ilk aşaması cihaz ile socket bağlantısını yapmaktır. Socket bağlantısı ile Eye Tribe izleme cihazını aktif ediyoruz.

Cihaz ile başarılı bir bağlantı gerçekleştirdikten sonra verilerin sürekli olarak alınması için geri dönüş parametresi içermeyen bir metod çağırıyoruz.

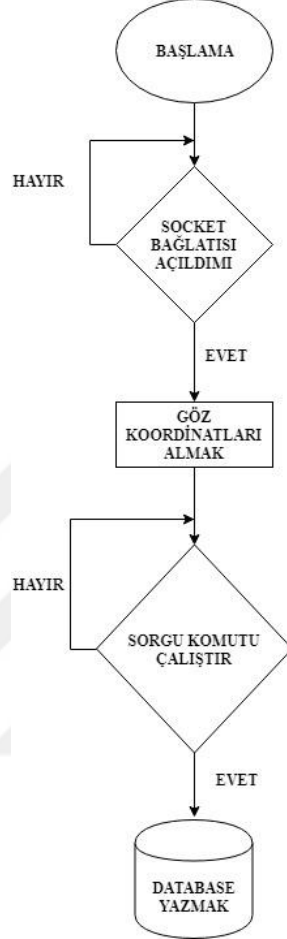
Eye Tribe izleme cihazından gelen verilerin yazılması için geri dönüş içermeyen ikinci bir metod yazılarak datanın sürekliliği sağlanmaktadır.

Son olarak gelen datalardan dairesel resimlerin çıkarılması için System.IO kütüphanesinden yararlanarak gerekli çizim için olan sınıflar sisteme implemente edilerek asenkron çalışan iki adet parametre alan bir fonksiyon yazıyoruz.

Alınan verilerin bir daire olabilmesi için son aşamada geri dönüşü double türünde olan içinde tek bir double türünde parametre alan fonksiyon tanımlaması yapılmıştır.

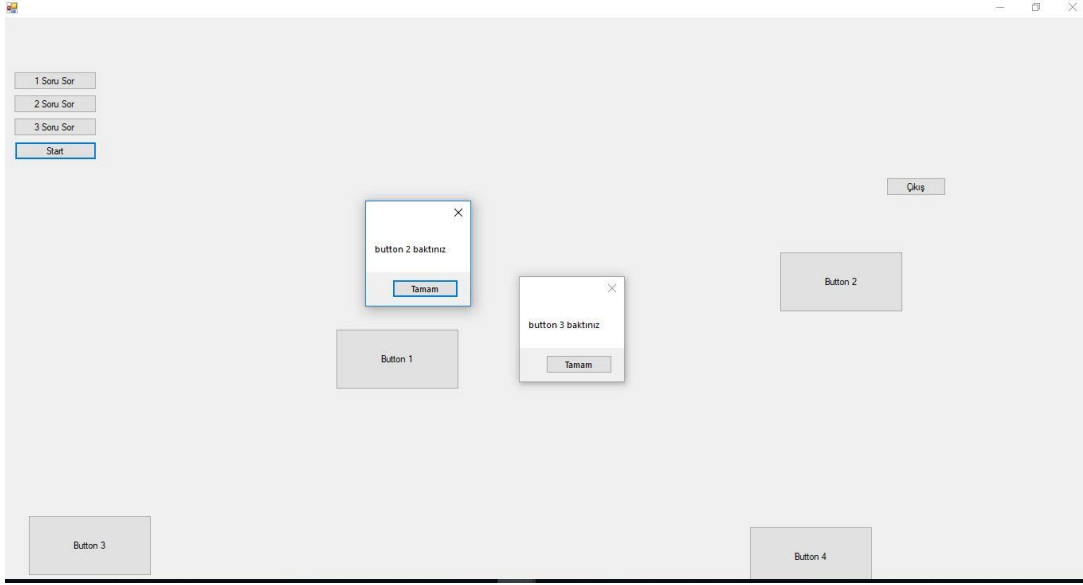
5.4 EYE TRİBE İZLEYİCİ İLE SORU CEVAP UYGULAMASI

Eye Tribe İzleyici ile bilgisayar ekranında çıkan sorulara kullanıcıların odaklanma ve gözleri ile verdikleri tepkiler analiz edilmiştir. Şekil 5.10'te programın akış diagramı yer almaktadır.



Şekil 5.10 : Soru Cevap Algoritması Akış Diagramı

Şekil 5.11'da programın görüntüsü yer almaktadır. Program ilk aşamada start düğmesine basılarak aktif edilir.



Şekil 5.11 : Program Ekran Görüntüsü

Uygulamamız C# ortamında ve windows platformunda çalışacak şekilde geliştirilmiştir. Start düğmesine basıldığında sistemde parametre almayan metod çağrılır. Kullanıcıya ilk soru gönderimi yapılır.

Calismak metodu içerisinde yer alan Connect metodu ile Eye Tribe İzleyiciye bağlantı sağlanır. Bağlantı işlemi başarılı olduğu zaman ListenerLoop metod geçilir ve bu metod ile sürekli olarak verinin alınması sağlanmaktadır. ListenerLoop metodu içinde ekranda sorulan sorulara karşılık gelen butonların X,X1 ve Y,Y1 kordinarları yer almaktadır. Bu koordinatlar cihaz tarafından gönderilen X,X1 ve Y,Y1 konumları arasından olduğu anda database ilgili kayıt metodu ile bilgiler kaydedilmektedir. Kayıt işlemi başarı ile devam ettiği sürece sıra ile ListenerLoop2 ve ListenerLoop3 metotları çalışacaktır.

Aşağıda yer alan veriler sistemde buton koordinatları eşleştirilmesi ile elde edilmiştir. Düzlem olarak X,X1 buton yatay konumda başlangıç ve bitişini Y,Y1 dikey konumda buton başlangıç ve bitiş noktasını temsil edilmiştir. Göz izleme eye-tribe cihazı ile göz ile baktığı nokta koordinatı tespit edilerek buton için belirlenen koordinat aralığında eşitleme işlemi yapılmaktadır. Testlerden alınan sonuçlar Tablo 5.1 ve Tablo 5.2’te gösterilmektedir.

Tablo 5.1: Örnek Uygulama Sonuçları

button 1 baktınız	2018-04-16 08:53:11.200	Buton 1 Bakınız
button 2 baktınız	2018-04-16 08:53:45.150	Buton 2 Bakınız
button 3 baktınız	2018-04-16 08:54:07.376	Buton 3 Bakınız
button 1 baktınız	2018-04-16 08:54:13.628	Buton 1 Bakınız
button 2 baktınız	2018-04-16 08:54:53.335	Buton 2 Bakınız
button 3 baktınız	2018-04-16 08:54:59.789	Buton 3 Bakınız
button 1 baktınız	2018-04-16 08:55:06.163	Buton 1 Bakınız
button 2 baktınız	2018-04-16 08:55:11.123	Buton 2 Bakınız
button 3 baktınız	2018-04-16 08:55:45.518	Buton 3 Bakınız
button 1 baktınız	2018-04-16 09:57:33.979	Buton 1 Bakınız
button 1 baktınız	2018-04-16 09:57:37.109	Buton 1 Bakınız
button 2 baktınız	2018-04-16 09:58:46.678	Buton 2 Bakınız
button 3 baktınız	2018-04-16 09:59:05.016	Buton 3 Bakınız
button 1 baktınız	2018-04-16 09:59:10.717	Buton 1 Bakınız
button 1 baktınız	2018-04-16 10:01:32.096	Buton 1 Bakınız
button 1 baktınız	2018-04-16 10:01:34.775	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:01:45.341	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:01:58.911	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:02:12.342	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:02:12.210	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:02:36.495	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:09:53.213	Buton 1 Bakınız
button 1 baktınız	2018-04-16 10:09:56.601	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:10:00.153	Buton 2 Bakınız
button 2 baktınız	2018-04-16 10:10:01.602	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:10:10.382	Buton 3 Bakınız
button 3 baktınız	2018-04-16 10:10:12.571	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:10:35.046	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:10:40.638	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:10:45.749	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:11:20.327	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:11:26.716	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:11:33.834	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:11:53.903	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:12:00.703	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:12:19.958	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:12:50.833	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:12:55.679	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:13:03.573	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:22:21.910	Buton 1 Bakınız

button 2 baktınız	2018-04-16 10:22:26.377	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:22:36.257	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:23:02.659	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:23:07.579	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:23:13.347	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:23:22.707	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:23:27.201	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:23:31.714	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:23:42.081	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:23:46.060	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:23:51.816	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:24:03.385	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:24:07.611	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:24:15.989	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:24:27.245	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:24:31.453	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:24:35.773	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:24:46.963	Buton 1 Bakınız
button 2 baktınız	2018-04-16 10:24:51.127	Buton 2 Bakınız
button 3 baktınız	2018-04-16 10:24:55.626	Buton 3 Bakınız
button 1 baktınız	2018-04-16 10:40:08.151	Buton 1 Bakınız
button 2 baktınız	2018-04-16 16:45:51.385	Buton 2 Bakınız
button 3 baktınız	2018-04-16 16:45:56.512	Buton 3 Bakınız
button 1 baktınız	2018-04-16 16:53:55.940	Buton 1 Bakınız
button 2 baktınız	2018-04-16 16:54:00.852	Buton 2 Bakınız
button 3 baktınız	2018-04-16 16:54:09.094	Buton 3 Bakınız

Tablo 5.2: Örnek Uygulama Sonuçları

Buton 1 Bakınız	2018-04-16 08:53:11.199
Buton 2 Bakınız	2018-04-16 08:53:45.146
Buton 3 Bakınız	2018-04-16 08:54:07.369
Buton 1 Bakınız	2018-04-16 08:54:13.625
Buton 2 Bakınız	2018-04-16 08:54:53.331
Buton 3 Bakınız	2018-04-16 08:54:59.787
Buton 1 Bakınız	2018-04-16 08:55:06.162
Buton 2 Bakınız	2018-04-16 08:55:11.122
Buton 3 Bakınız	2018-04-16 08:55:45.474
Buton 1 Bakınız	2018-04-16 09:57:33.945
Buton 1 Bakınız	2018-04-16 09:57:37.109
Buton 2 Bakınız	2018-04-16 09:58:46.676
Buton 3 Bakınız	2018-04-16 09:59:05.013
Buton 1 Bakınız	2018-04-16 09:59:10.716
Buton 1 Bakınız	2018-04-16 10:01:32.064
Buton 1 Bakınız	2018-04-16 10:01:34.774
Buton 2 Bakınız	2018-04-16 10:01:45.340
Buton 3 Bakınız	2018-04-16 10:01:58.908
Buton 1 Bakınız	2018-04-16 10:02:12.341
Buton 2 Bakınız	2018-04-16 10:02:24.942
Buton 3 Bakınız	2018-04-16 10:02:36.492
Buton 1 Bakınız	2018-04-16 10:09:53.180
Buton 1 Bakınız	2018-04-16 10:09:56.598
Buton 2 Bakınız	2018-04-16 10:10:00.150
Buton 2 Bakınız	2018-04-16 10:10:01.599
Buton 3 Bakınız	2018-04-16 10:10:10.380
Buton 3 Bakınız	2018-04-16 10:10:12.568
Buton 1 Bakınız	2018-04-16 10:10:35.044
Buton 2 Bakınız	2018-04-16 10:10:40.636
Buton 3 Bakınız	2018-04-16 10:10:45.749
Buton 1 Bakınız	2018-04-16 10:11:20.325
Buton 2 Bakınız	2018-04-16 10:11:26.715
Buton 3 Bakınız	2018-04-16 10:11:33.832
Buton 1 Bakınız	2018-04-16 10:11:53.901
Buton 2 Bakınız	2018-04-16 10:12:00.700
Buton 3 Bakınız	2018-04-16 10:12:19.956
Buton 1 Bakınız	2018-04-16 10:12:50.828

Buton 2 Bakınız	2018-04-16 10:12:55.676
Buton 3 Bakınız	2018-04-16 10:13:03.572
Buton 1 Bakınız	2018-04-16 10:22:21.879
Buton 2 Bakınız	2018-04-16 10:22:26.375
Buton 3 Bakınız	2018-04-16 10:22:36.254
Buton 1 Bakınız	2018-04-16 10:23:02.655
Buton 2 Bakınız	2018-04-16 10:23:07.574
Buton 3 Bakınız	2018-04-16 10:23:13.343
Buton 1 Bakınız	2018-04-16 10:23:22.702
Buton 2 Bakınız	2018-04-16 10:23:27.199
Buton 3 Bakınız	2018-04-16 10:23:31.713
Buton 1 Bakınız	2018-04-16 10:23:42.079
Buton 2 Bakınız	2018-04-16 10:23:46.056
Buton 3 Bakınız	2018-04-16 10:23:51.814
Buton 1 Bakınız	2018-04-16 10:24:03.382
Buton 2 Bakınız	2018-04-16 10:24:07.607
Buton 3 Bakınız	2018-04-16 10:24:15.982
Buton 1 Bakınız	2018-04-16 10:24:27.240
Buton 2 Bakınız	2018-04-16 10:24:31.448
Buton 3 Bakınız	2018-04-16 10:24:35.751
Buton 1 Bakınız	2018-04-16 10:24:46.959
Buton 2 Bakınız	2018-04-16 10:24:51.103
Buton 3 Bakınız	2018-04-16 10:24:55.623
Buton 1 Bakınız	2018-04-16 10:40:08.149
Buton 2 Bakınız	2018-04-16 16:45:51.383
Buton 3 Bakınız	2018-04-16 16:45:56.505
Buton 1 Bakınız	2018-04-16 16:53:55.938
Buton 2 Bakınız	2018-04-16 16:54:00.849
Buton 3 Bakınız	2018-04-16 16:54:09.090

Çalışmamızda bir butona bakmanın ortalama 11 mili saniye olduğu tespit edilmiştir. Tablo 5.3’de hesaplanmış hali listelenmiştir.

Tablo 5.3: Örnek Uygulama Zamanı Sonuçları

Cevap	Soru	Süre
16.04.2018 08:53:11, 200	16.04.2018 08:53:11,199	00:00:00,001
16.04.2018 08:53:45,150	16.04.2018 08:53:45,146	00:00:00,004
16.04.2018 08:54:07,376	16.04.2018 08:54:07,369	00:00:00,007
16.04.2018 08:54:13,628	16.04.2018 08:54:13,625	00:00:00,003
16.04.2018 08:54:53,335	16.04.2018 08:54:53,331	00:00:00,004
16.04.2018 08:54:59,789	16.04.2018 08:54:59,787	00:00:00,002
16.04.2018 08:55:06,163	16.04.2018 08:55:06,162	00:00:00,001
16.04.2018 08:55:11,123	16.04.2018 08:55:11,122	00:00:00,001
16.04.2018 08:55:45,518	16.04.2018 08:55:45,474	00:00:00,044
16.04.2018 09:57:33,979	16.04.2018 09:57:33,945	00:00:00,034
16.04.2018 09:57:37,109	16.04.2018 09:57:37,109	00:00:00,000
16.04.2018 09:58:46,678	16.04.2018 09:58:46,676	00:00:00,002
16.04.2018 09:59:05,016	16.04.2018 09:59:05,013	00:00:00,003
16.04.2018 09:59:10,717	16.04.2018 09:59:10,716	00:00:00,001
16.04.2018 10:01:32,096	16.04.2018 10:01:32,064	00:00:00,032
16.04.2018 10:01:34,775	16.04.2018 10:01:34,774	00:00:00,001
16.04.2018 10:01:45,341	16.04.2018 10:01:45,340	00:00:00,001
16.04.2018 10:01:58,911	16.04.2018 10:01:58,908	00:00:00,003
16.04.2018 10:02:12,342	16.04.2018 10:02:12,341	00:00:00,001
16.04.2018 10:02:12, 210	16.04.2018 10:02:12,000	00:00:00, 210
16.04.2018 10:02:36,495	16.04.2018 10:02:36,492	00:00:00,003
16.04.2018 10:09:53, 213	16.04.2018 10:09:53,180	00:00:00,033
16.04.2018 10:09:56,601	16.04.2018 10:09:56,598	00:00:00,003
16.04.2018 10:10:00,153	16.04.2018 10:10:00,150	00:00:00,003
16.04.2018 10:10:01,602	16.04.2018 10:10:01,599	00:00:00,003
16.04.2018 10:10:10,382	16.04.2018 10:10:10,380	00:00:00,002
16.04.2018 10:10:12,571	16.04.2018 10:10:12,568	00:00:00,003
16.04.2018 10:10:35,046	16.04.2018 10:10:35,044	00:00:00,002
16.04.2018 10:10:40,638	16.04.2018 10:10:40,636	00:00:00,002
16.04.2018 10:10:45,749	16.04.2018 10:10:45,749	00:00:00,000
16.04.2018 10:11:20,327	16.04.2018 10:11:20,325	00:00:00,002
16.04.2018 10:11:26,716	16.04.2018 10:11:26,715	00:00:00,001
16.04.2018 10:11:33,834	16.04.2018 10:11:33,832	00:00:00,002
16.04.2018 10:11:53,903	16.04.2018 10:11:53,901	00:00:00,002
16.04.2018 10:12:00,703	16.04.2018 10:12:00,700	00:00:00,003
16.04.2018 10:12:19,958	16.04.2018 10:12:19,956	00:00:00,002
16.04.2018 10:12:50,833	16.04.2018 10:12:50,828	00:00:00,005
16.04.2018 10:12:55,679	16.04.2018 10:12:55,676	00:00:00,003
16.04.2018 10:13:03,573	16.04.2018 10:13:03,572	00:00:00,001
16.04.2018 10:22:21,910	16.04.2018 10:22:21,879	00:00:00,031
16.04.2018 10:22:26,377	16.04.2018 10:22:26,375	00:00:00,002
16.04.2018 10:22:36, 257	16.04.2018 10:22:36, 254	00:00:00,003
16.04.2018 10:23:02,659	16.04.2018 10:23:02,655	00:00:00,004

16.04.2018 10:23:07,579	16.04.2018 10:23:07,574	00:00:00,005
16.04.2018 10:23:13,347	16.04.2018 10:23:13,343	00:00:00,004
16.04.2018 10:23:22,707	16.04.2018 10:23:22,702	00:00:00,005
16.04.2018 10:23:27, 201	16.04.2018 10:23:27,199	00:00:00,002
Cevap	Soru	Süre
16.04.2018 10:23:31,714	16.04.2018 10:23:31,713	00:00:00,001
16.04.2018 10:23:42,081	16.04.2018 10:23:42,079	00:00:00,002
16.04.2018 10:23:46,060	16.04.2018 10:23:46,056	00:00:00,004
16.04.2018 10:23:51,816	16.04.2018 10:23:51,814	00:00:00,002
16.04.2018 10:24:03,385	16.04.2018 10:24:03,382	00:00:00,003
16.04.2018 10:24:07,611	16.04.2018 10:24:07,607	00:00:00,004
16.04.2018 10:24:15,989	16.04.2018 10:24:15,982	00:00:00,007
16.04.2018 10:24:27, 245	16.04.2018 10:24:27, 240	00:00:00,005
16.04.2018 10:24:31,453	16.04.2018 10:24:31,448	00:00:00,005
16.04.2018 10:24:35,773	16.04.2018 10:24:35,751	00:00:00,022
16.04.2018 10:24:46,963	16.04.2018 10:24:46,959	00:00:00,004
16.04.2018 10:24:51,127	16.04.2018 10:24:51,103	00:00:00,024
16.04.2018 10:24:55,626	16.04.2018 10:24:55,623	00:00:00,003
16.04.2018 10:40:08,151	16.04.2018 10:40:08,149	00:00:00,002
16.04.2018 16:45:51,385	16.04.2018 16:45:51,383	00:00:00,002
16.04.2018 16:45:56,512	16.04.2018 16:45:56,505	00:00:00,007
16.04.2018 16:53:55,940	16.04.2018 16:53:55,938	00:00:00,002
16.04.2018 16:54:00,852	16.04.2018 16:54:00,849	00:00:00,003
16.04.2018 16:54:09,094	16.04.2018 16:54:09,090	00:00:00,004
10.02.2019 19:51:39,622	10.02.2019 19:51:39,525	00:00:00,097
10.02.2019 19:51:50, 234	10.02.2019 19:51:50, 203	00:00:00,031
Toplam Ortalama Tıklama süresi	00:00:00,011	
Standart Sapması	00:00:00,029	

5.4.1. Database ve Tablo Yapımız

MSSQL (Microsoft – Structured Query Language) veri tabanı sunucu yazılım türlerinden biridir. Bu platform sayesinde, veri tabanı oluşturma ve geliştirme çok kolay bir hal almıştır. MS SQL (Microsoft – Structured Query Language)’in database altyapısı ile online veya online olmayan işlemlerimiz için ilişkisel veri tabanları oluşturabilir. Oluşturduğumuz bu veri tabanları üzerinde tablolara oluşturarak datalarımızı kolay bir şekilde yöneterek (Ekleme, Güncelleme, Silme, Listeleme v.b.) güvenli bir şekilde saklayabiliriz.

Uygulamamızda yer alan database table örneklerii Şekil 5.12 ve Şekil 5.13’te gösterilmektedir.

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
soru	nvarchar(255)	<input checked="" type="checkbox"/>
tarih	nvarchar(255)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

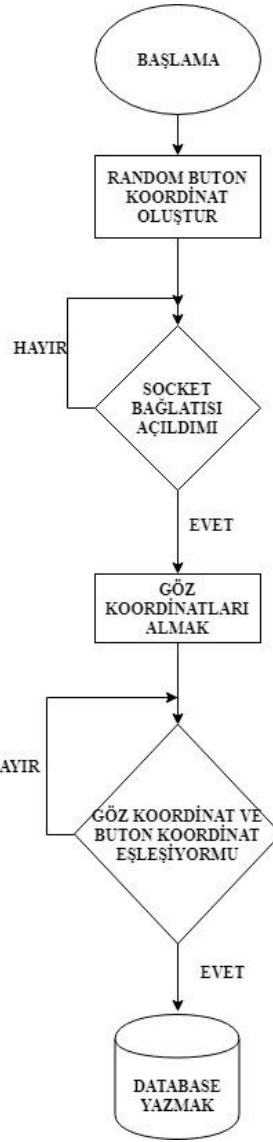
Şekil 5.12 : Soru Database Şeması

Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
buttonmesaj	nvarchar(255)	<input checked="" type="checkbox"/>
buttonsuresi	nvarchar(255)	<input checked="" type="checkbox"/>
soru	nvarchar(255)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil 5.13 : Cevap Database Şeması

5.5 EYE TRİBE İZLEYİCİ İLE OTOMATİK SORU CEVAP UYGULAMASI

Eye Tribe İzleyici ile bilgisayar ekranında otomatik olarak üretilen farklı ve sabit olmayan koordinatlarda çıkan sorulara karşı kullanıcıların odaklanma ve gözleri ile verdikleri tepkiler analiz edilmiştir. Şekil 5.14'da programın akış diagramı yer almaktadır.



Şekil 5.14 : Otomatik Soru Cevap Akış Diagramı

Uygulamamız çalışma sırasında ilk önce timer nesnesi ile beraber random bir buton oluşturulur. Oluşturulan buton nesnesinin koordinatları Point nesnesi aracılığı ile elde edilir.

Eye Tribe İzleyici başlatma komutu kod kısmında çalıştırılır. Sürekli olarak veri akışı sağlanması için cihaza Send isiminde ve parametre olarak string türünde veri gönderilir. ListenerLoop ile Eye Tribe cihazından sürekli olarak koordinatlar alınmaktadır. Alınan bu koordinatlar metod içinde buton koordinatları ile karşılaştırılmakta ve aynı koordinat düzeni ile keşirse sistem otomatik olarak tıklanır ve integer türünde parametre alan fonksiyon çağrılmaktadır.

Bu fonksiyon ile database kayıt yapılmaktadır. Sistem uyarı olarak başarılı bir şekilde eşleştirme yaparsa windows sisteminin default olan uyarı sesini verdirmekteyiz. Oluşturulan uygulamadan elde edilen sonuçlar Tablo 5.4’de gösterilmektedir.

Tablo 5.4: Örnek Uygulama Sonuçları

SORU	CEVAP	FARK
25.11.2018 13:11:49,788	25.11.2018 13:11:52,735	00:00:02,947
25.11.2018 13:11:52,880	25.11.2018 13:11:55,828	00:00:02,948
25.11.2018 13:11:55,836	25.11.2018 13:11:58,832	00:00:02,996
25.11.2018 13:11:59,521	25.11.2018 13:12:02,469	00:00:02,948
25.11.2018 13:12:02,524	25.11.2018 13:12:05,527	00:00:03,003
25.11.2018 13:12:05,528	25.11.2018 13:12:05,529	00:00:00,001
25.11.2018 13:12:05,530	25.11.2018 13:12:08,530	00:00:03,000
25.11.2018 13:12:08,976	25.11.2018 13:12:11,915	00:00:02,939
25.11.2018 13:12:11,997	25.11.2018 13:12:15,054	00:00:03,057
25.11.2018 13:12:15,071	25.11.2018 13:12:18,062	00:00:02,991
25.11.2018 13:12:18,162	25.11.2018 13:12:21,114	00:00:02,952
25.11.2018 13:12:21,116	25.11.2018 13:12:21,164	00:00:00,048
25.11.2018 13:12:21,165	25.11.2018 13:12:24,167	00:00:03,002
25.11.2018 13:12:24,169	25.11.2018 13:12:27,171	00:00:03,002
25.11.2018 13:12:27,365	25.11.2018 13:12:30,178	00:00:02,813
25.11.2018 13:12:30,179	25.11.2018 13:12:30,367	00:00:00,188
25.11.2018 13:12:30,368	25.11.2018 13:12:33,369	00:00:03,001
25.11.2018 13:12:33,376	25.11.2018 13:12:36,378	00:00:03,002
25.11.2018 13:12:39,412	25.11.2018 13:12:42,440	00:00:03,028
25.11.2018 13:12:42,443	25.11.2018 13:12:45,445	00:00:03,002
25.11.2018 13:12:45,551	25.11.2018 13:12:48,450	00:00:02,899
25.11.2018 13:12:48,451	25.11.2018 13:12:48,553	00:00:00,102
25.11.2018 13:12:48,643	25.11.2018 13:12:51,646	00:00:03,003
25.11.2018 13:12:51,705	25.11.2018 13:12:51,708	00:00:00,003
25.11.2018 13:16:25,600	25.11.2018 13:16:28,543	00:00:02,943
25.11.2018 13:16:28,631	25.11.2018 13:16:31,634	00:00:03,003
25.11.2018 13:16:31,639	25.11.2018 13:16:31,711	00:00:00,072
25.11.2018 13:16:31,716	25.11.2018 13:16:34,716	00:00:03,000
25.11.2018 13:16:34,745	25.11.2018 13:16:37,747	00:00:03,002
25.11.2018 13:16:37,774	25.11.2018 13:16:40,752	00:00:02,978
25.11.2018 13:16:40,753	25.11.2018 13:16:40,776	00:00:00,023
25.11.2018 13:16:40,777	25.11.2018 13:16:43,779	00:00:03,002
25.11.2018 13:16:43,893	25.11.2018 13:16:46,787	00:00:02,894
25.11.2018 13:16:46,789	25.11.2018 13:16:46,895	00:00:00,106
25.11.2018 13:16:46,896	25.11.2018 13:16:49,898	00:00:03,002
25.11.2018 13:16:50,010	25.11.2018 13:16:52,932	00:00:02,922
25.11.2018 13:16:52,935	25.11.2018 13:16:53,012	00:00:00,077

25.11.2018 13:16:53,073	25.11.2018 13:16:56,016	00:00:02,943
25.11.2018 13:16:56,018	25.11.2018 13:16:56,077	00:00:00,059
25.11.2018 13:16:56,136	25.11.2018 13:16:59,080	00:00:02,944
25.11.2018 13:16:59,081	25.11.2018 13:16:59,138	00:00:00,057
25.11.2018 13:17:02, 208	25.11.2018 13:17:05,149	00:00:02,941
25.11.2018 13:17:05,151	25.11.2018 13:17:08,154	00:00:03,003
25.11.2018 13:17:08,156	25.11.2018 13:17:11,158	00:00:03,002
25.11.2018 13:17:11,309	25.11.2018 13:17:14,165	00:00:02,856
25.11.2018 13:17:14,167	25.11.2018 13:17:14,311	00:00:00,144
25.11.2018 13:17:14,312	25.11.2018 13:17:17,314	00:00:03,002
25.11.2018 13:17:17,482	25.11.2018 13:17:20,326	00:00:02,844
25.11.2018 13:17:20,523	25.11.2018 13:17:23,487	00:00:02,964
25.11.2018 13:17:23,528	25.11.2018 13:17:26,528	00:00:03,000
25.11.2018 13:17:26,663	25.11.2018 13:17:29,587	00:00:02,924
25.11.2018 13:17:29,590	25.11.2018 13:17:32,591	00:00:03,001
25.11.2018 13:17:32,593	25.11.2018 13:17:35,595	00:00:03,002
25.11.2018 13:17:38,680	25.11.2018 13:17:41,646	00:00:02,966
25.11.2018 13:17:41,669	25.11.2018 13:17:44,652	00:00:02,983
25.11.2018 13:17:44,671	25.11.2018 13:17:47,675	00:00:03,004
25.11.2018 13:17:47,681	25.11.2018 13:17:50,683	00:00:03,002
25.11.2018 13:17:57,632	25.11.2018 13:18:00,573	00:00:02,941
25.11.2018 13:18:00,574	25.11.2018 13:18:03,577	00:00:03,003
25.11.2018 13:18:03,580	25.11.2018 13:18:06,581	00:00:03,001
25.11.2018 13:18:06,589	25.11.2018 13:18:06,593	00:00:00,004
22.12.2018 20:21:42,901	22.12.2018 20:21:45,843	00:00:02,942
22.12.2018 20:21:45,994	22.12.2018 20:21:48,934	00:00:02,940
22.12.2018 20:21:49,102	22.12.2018 20:21:52,043	00:00:02,941
22.12.2018 20:21:52,134	22.12.2018 20:21:55,136	00:00:03,002
22.12.2018 20:21:58, 229	22.12.2018 20:22:01,168	00:00:02,939
22.12.2018 20:22:01,175	22.12.2018 20:22:04,172	00:00:02,997
22.12.2018 20:22:04, 211	22.12.2018 20:22:07, 217	00:00:03,006
22.12.2018 20:22:07, 235	22.12.2018 20:22:10, 222	00:00:02,987
22.12.2018 20:22:10, 223	22.12.2018 20:22:10, 237	00:00:00,014
22.12.2018 20:22:10,301	22.12.2018 20:22:13, 259	00:00:02,958
22.12.2018 20:22:13, 261	22.12.2018 20:22:13,303	00:00:00,042
22.12.2018 20:22:13,304	22.12.2018 20:22:16,306	00:00:03,002
22.12.2018 20:22:16,399	22.12.2018 20:22:19,310	00:00:02,911
22.12.2018 20:22:19,312	22.12.2018 20:22:19,401	00:00:00,089
22.12.2018 20:22:19,466	22.12.2018 20:22:22,405	00:00:02,939
22.12.2018 20:22:22,469	22.12.2018 20:22:25,471	00:00:03,002
22.12.2018 20:22:25,542	22.12.2018 20:22:28,477	00:00:02,935
22.12.2018 20:22:28,480	22.12.2018 20:22:28,545	00:00:00,065
22.12.2018 20:22:28,615	22.12.2018 20:22:31,619	00:00:03,004
22.12.2018 20:22:31,698	22.12.2018 20:22:34,638	00:00:02,940

22.12.2018 20:22:34,659	22.12.2018 20:22:37,643	00:00:02,984
22.12.2018 20:22:37,645	22.12.2018 20:22:37,661	00:00:00,016
22.12.2018 20:22:37,732	22.12.2018 20:22:40,666	00:00:02,934
22.12.2018 20:22:40,677	22.12.2018 20:22:40,734	00:00:00,057
22.12.2018 20:22:40,735	22.12.2018 20:22:43,737	00:00:03,002
22.12.2018 20:22:43,879	22.12.2018 20:22:46,743	00:00:02,864
22.12.2018 20:22:46,746	22.12.2018 20:22:46,883	00:00:00,137
22.12.2018 20:22:46,987	22.12.2018 20:22:49,887	00:00:02,900
22.12.2018 20:22:49,965	22.12.2018 20:22:49,989	00:00:00,024
22.12.2018 20:26:01,806	22.12.2018 20:26:04,734	00:00:02,928
22.12.2018 20:26:08,302	22.12.2018 20:26:11,304	00:00:03,002
22.12.2018 20:26:15,525	22.12.2018 20:26:16,073	00:00:00,548
22.12.2018 20:26:23,884	22.12.2018 20:26:26,827	00:00:02,943
22.12.2018 20:26:30,387	22.12.2018 20:26:33,390	00:00:03,003
22.12.2018 20:26:35,161	22.12.2018 20:26:38,163	00:00:03,002
22.12.2018 20:28:09,759	22.12.2018 20:28:12,702	00:00:02,943
22.12.2018 20:28:19,269	22.12.2018 20:28:20,527	00:00:01,258
22.12.2018 20:28:22,300	22.12.2018 20:28:25,297	00:00:02,997
22.12.2018 20:28:28,841	22.12.2018 20:28:31,845	00:00:03,004
22.12.2018 20:28:33,615	22.12.2018 20:28:36,618	00:00:03,003
22.12.2018 20:28:43,182	22.12.2018 20:28:44,474	00:00:01,292
22.12.2018 20:28:51,014	22.12.2018 20:28:52,910	00:00:01,896
22.12.2018 20:28:59,451	22.12.2018 20:29:00,700	00:00:01,249
22.12.2018 20:29:02,468	22.12.2018 20:29:05,470	00:00:03,002
22.12.2018 20:29:07,237	22.12.2018 20:29:10,238	00:00:03,001
22.12.2018 20:29:16,745	22.12.2018 20:29:16,796	00:00:00,051
22.12.2018 20:32:32,119	22.12.2018 20:32:35,061	00:00:02,942
22.12.2018 20:32:36,855	22.12.2018 20:32:39,854	00:00:02,999
22.12.2018 20:32:43,414	22.12.2018 20:32:46,433	00:00:03,019
22.12.2018 20:32:48,775	22.12.2018 20:32:51,208	00:00:02,433
22.12.2018 20:35:59,667	22.12.2018 20:36:02,697	00:00:03,030
22.12.2018 20:35:59,727	22.12.2018 20:36:02,730	00:00:03,003
22.12.2018 20:36:04,558	22.12.2018 20:36:09,282	00:00:04,724
22.12.2018 20:36:06,277	22.12.2018 20:36:14,063	00:00:07,786
22.12.2018 20:36:11,056	22.12.2018 20:36:18,839	00:00:07,783
22.12.2018 20:36:16,916	22.12.2018 20:36:25,394	00:00:08,478
22.12.2018 20:36:25,369	22.12.2018 20:39:00,795	00:02:35,426
22.12.2018 20:38:57,854	22.12.2018 20:39:07,543	00:00:09,689
	Ortalama Süre	00:00:02,456
	Standart Sapma	00:00:14,106

Genel olarak mikro saniye bazında veya dış koşullara bağlı olarak maksimum 3 saniyeyi geçmeden çıkan butonların hepsine tıklanma işlemi gerçekleştirilmiştir. Tablo yapımız Şekil 5.15 gösterilmektedir.

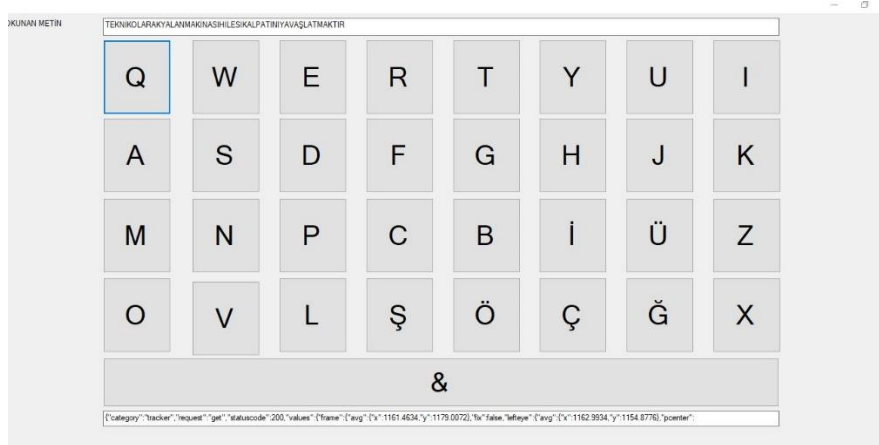
Column Name	Data Type	Allow Nulls
id	int	<input type="checkbox"/>
buton_zaman	nvarchar(250)	<input checked="" type="checkbox"/>
buton_aciklama	nvarchar(250)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil 5.15 : Otomatik Soru Cevap Tablo Yapısı

5.6 EYE TRİBE İZLEYİCİ İLE SANAL KLAVYE UYGULAMASI

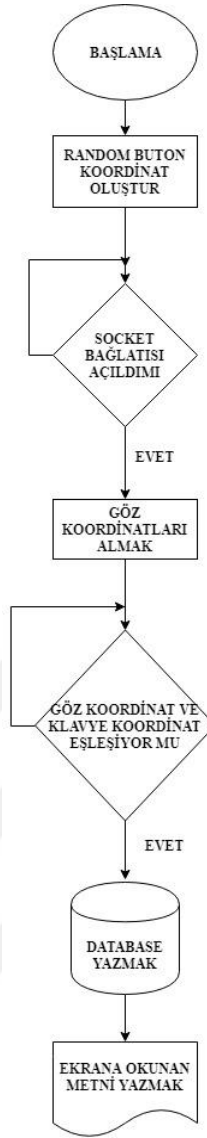
Eye Tribe İzleyici ile bilgisayar ekranına klavye özelliğini kazandırmak kurgulanmıştır. Amacımız bu klavye yöntemi ile belirtilen kelime ve cümleleri ne kadar sürede yazılacağını tespit etmek amaçlanmıştır. Seçilen cümleler veya kelimeler rassal olarak sitelerden veya ikinci bir kişinin okuması ile belirlenen metinlerdir.

Ekran üstünde yer alan her harfin koordinatları belirlenmiştir. Yazılan metinlerin süre tespiti için okunan her harf database kaydedilmektedir. Uygulamanın Şekil 5.16'de ekran görüntüsü yer almaktadır. Yazılan her bir harften sonra sistem ses ile uyarı vermektedir. Uygulama süreçleri kendi tarafımdan kurgulanmış ve yazılmıştır.



Şekil 5.16 : Sanal Klavye Ekran Görüntüsü

Uygulamanın akış diyagramı Şekil 5.17'de gösterilmektedir.



Şekil 5.17 : Sanal Klavye Uygulaması Akış Diagramı

Sanal klavye uygulaması başlatıldığı yapıcı metod tüm harflerin koordinatları oluşturulur. Form nesnesi oluşturulduğu anda timer zamanlayıcı devreye girer. Timer zamanlayıcı nesnesi içinde Eye Tribe izleyici bağlantı fonksiyonu çağrılır. Socket bağlantı başarılı olduğu anda sisteme true değer dönderir.

ListenerLoop metodu çağrılarak verileri devamlı akışı sağlar. ListenerLoop metodu içerisinde sanal klavyede kullanıcı hangi harfe odaklanmışsa o odaklanmış olduğu harfin koordinat sisteminde yeri tespit edilir. Verilerin akışı çok hızlı olduğundan harfler yakalandıktan sonra bir sesli uyarı sistemi konularak uygulama bekleme konumuna alınmaktadır. Ayrıca ard arda aynı harfin database yazılmaması için koleksiyon nesnesi tanımlanır. Bu koleksiyon nesnesine eklenen her harf için bir öncesinde kendisi ile aynı olup olmadığı kontrol edilir. Aynıısı var ise kolaksiyon nesnesine harf eklenmez.

Butonların X ,X1 ve Y,Y1 koordinatlarının oluşturulması için bir metod daha yazılmıştır. Uygulamada okutulan ve database yazılan cümleler Tablo 5.5’de gösterilmektedir.

Tablo 5.5: Örnek Kelime Oluşturma Uygulaması

B	2018-06-04 17:56:08.656
O	2018-06-04 17:56:11.655
R	2018-06-04 17:56:14.655
S	2018-06-04 17:56:17.657
A	2018-06-04 17:56:20.655
D	2018-06-04 17:56:23.641
A	2018-06-04 17:56:26.656
İ	2018-06-04 17:56:29.641
Ş	2018-06-04 17:56:32.643
L	2018-06-04 17:56:35.641
E	2018-06-04 17:56:38.640
M	2018-06-04 17:56:41.672
Y	2018-06-04 17:56:44.657
A	2018-06-04 17:56:47.655
P	2018-06-04 17:56:59.654
A	2018-06-04 17:57:02.654
N	2018-06-04 17:57:05.642
A	2018-06-04 17:57:08.647
Y	2018-06-04 17:57:11.641
E	2018-06-04 17:57:14.642
T	2018-06-04 17:57:17.641
K	2018-06-04 17:57:20.641
İ	2018-06-04 17:57:26.655
B	2018-06-04 17:57:32.655
E	2018-06-04 17:57:35.656
L	2018-06-04 17:57:38.656
G	2018-06-04 17:57:41.656
E	2018-06-04 17:57:44.655
S	2018-06-04 17:57:47.641
İ	2018-06-04 17:57:50.640
Z	2018-06-04 17:57:53.642
O	2018-06-04 17:57:56.642
R	2018-06-04 17:57:59.642
U	2018-06-04 17:58:02.672

N	2018-06-04 17:58:08.656
L	2018-06-04 17:58:11.654
U	2018-06-04 17:58:17.657
O	2018-06-04 17:58:20.655
L	2018-06-04 17:58:35.640
D	2018-06-04 17:58:38.641
U	2018-06-04 17:58:41.642

Uygulamanın database yapısı Şekil 5.18’da gösterilmektedir. Bu yapıda cümleyi 01:54 sn’de yazılmıştır.

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Harf	nchar(1)	<input checked="" type="checkbox"/>
zaman	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Şekil 5.18 : Sanal Klavye Database Yapısı

6. SONUÇ VE TARTIŞMA

Bu tez çalışmasında, C# programlama dili vasıtasıyla göz hareketlerini izlemeye olanak sağlayan bir BBA yazılımı geliştirilmiştir. Bu sayede hareket yetisini katbetmiş insanlara göz izleme takibi yapılarak cümleler kurması amaçlanmıştır.

Bu amaçla, Eye Tribe izleyici ile ilk veri akışının elde edilmesinden, sanal klavye uygulamasına kadar bir çok deneyler yapılmıştır.

Sıcaklık uygulamasında kullanıcıların ekranda baktıkları yerleri canlı olarak nokta şeklinde küçük daireler çizdirilmiştir.

Soru cevap sisteminde kullanıcılara ekrandan sorular gösterilmiş ve bunlara verdikleri cevapların süreleri bir tablo şeklinde gösterilmiştir. Ortalama olarak en fazla 3 saniye içinde cevap verme hızları tespit edilmiştir.

Otomatik soru ve cevap verme sisteminde ise ekranda koordinat olarak rassal gösterilen butonlara kullanıcıların tıklama süreleri tespit edilmiştir. Ortalama olarak en fazla 3 saniye olarak kullanıcıların cevap verme hızları tespit edilmiştir.

Sanal klavye uygulamasında okutulan “zaman farkı hesaplama formülü derste anlatıldı bunu koda dökmek gerekicek debug modsa işlemi yavaşlatır” cümlesinde analiz yapılmıştır. Analiz sonuçları kelime bazında aşağıdaki Tablo 6.1’de gösterilmiştir.

Tablo 6.1: Örnek Cümle ve Analiz Sonuçları

	AD	HARF	BAŞLAMA SÜRESİ	BİTİŞ SÜRESİ	OKUNMA SÜRESİ
1	ZAMAN	5	10.02.2019 18:12:44,992	10.02.2019 18:12:57,000	00:00:12,008
2	FARKI	5	10.02.2019 18:12:59,993	10.02.2019 18:13:17,994	00:00:18,001
3	HESAPLAMA	9	10.02.2019 18:13:20,980	10.02.2019 18:14:14,980	00:00:54,000
4	FORMÜLÜ	7	10.02.2019 18:14:17,968	10.02.2019 18:14:50,980	00:00:33,012
5	DERSTE	6	10.02.2019 18:14:53,980	10.02.2019 18:15:23,988	00:00:30,008
6	ANLATILDI	9	10.02.2019 18:15:29,980	10.02.2019 18:16:02,983	00:00:33,003
7	BUNU	4	10.02.2019 18:16:08,981	10.02.2019 18:16:20,980	00:00:11,999
8	KODA	4	10.02.2019 18:16:23,983	10.02.2019 18:16:32,971	00:00:08,988
9	DÖKMEK	6	10.02.2019 18:16:29,000	10.02.2019 18:17:02,981	00:00:33,981
10	GEREKİCEK	9	10.02.2019 18:17:05,981	10.02.2019 18:17:32,982	00:00:27,001
11	DEBUG	5	10.02.2019 18:17:38,980	10.02.2019 18:17:50,964	00:00:11,984
12	MODSA	5	10.02.2019 18:17:53,963	10.02.2019 18:18:20,979	00:00:27,016
13	İŞLEMİ	6	10.02.2019 18:18:23,981	10.02.2019 18:19:08,974	00:00:44,993
14	YAVAŞLATIR	10	10.02.2019 18:19:11,979	10.02.2019 18:20:02,994	00:00:51,015
TOPLAM					00:06:37,009

Sanal klavye uygulamasında 5 kullanıcıya “borsada işlem yapana yetki belgesi zorunlu oldu” metin okutulmuş ve okutulan bu metinlerin analiz sonuçları Tablo 6.2’de gösterilmiştir.

Tablo 6.2: Örnek Cümle ve Analiz Sonuçları

ENGİN					
1	AD	HARF	BAŞLAMA SÜRESİ	BİTİŞ SÜRESİ	OKUNMA SÜRESİ
2	BORSADA	7	01.02.2019 14:47:51,419	01.02.2019 14:48:14,026	00:00:22,607
3	İŞLEM	5	01.02.2019 14:48:16,860	01.02.2019 14:48:42,401	00:00:25,541
4	YAPANA	6	01.02.2019 14:48:45,403	01.02.2019 14:49:00,386	00:00:14,983
5	YETKİ	5	01.02.2019 14:49:03,392	01.02.2019 14:49:24,402	00:00:21,010
6	BELGESİ	7	01.02.2019 14:49:33,402	01.02.2019 14:50:15,388	00:00:41,986
7	ZORUNLU	7	01.02.2019 14:50:18,417	01.02.2019 14:50:48,404	00:00:29,987
8	OLDU	4	01.02.2019 14:50:51,404	01.02.2019 14:51:06,416	00:00:15,012
				TOPLAM	00:02:51,126
BORA					
1	AD	HARF	BAŞLAMA SÜRESİ	BİTİŞ SÜRESİ	OKUNMA SÜRESİ
2	BORSADA	7	01.02.2019 14:52:03,978	01.02.2019 14:52:42,979	00:00:39,001
3	İŞLEM	5	01.02.2019 14:52:45,979	01.02.2019 14:53:21,979	00:00:36,000
4	YAPANA	6	01.02.2019 14:53:24,981	01.02.2019 14:53:51,995	00:00:27,014
5	YETKİ	5	01.02.2019 14:53:58,213	01.02.2019 14:54:13,217	00:00:15,004
6	BELGESİ	7	01.02.2019 14:54:19,199	01.02.2019 14:54:55,197	00:00:35,998
7	ZORUNLU	7	01.02.2019 14:55:01,198	01.02.2019 14:55:28,214	00:00:27,016
8	OLDU	4	01.02.2019 14:55:40,200	01.02.2019 14:55:55,229	00:00:15,029
				TOPLAM	00:03:15,062
ORÇUN					
1	AD	HARF	BAŞLAMA SÜRESİ	BİTİŞ SÜRESİ	OKUNMA SÜRESİ
2	BORSADA	7	01.02.2019 15:13:27,699	01.02.2019 15:13:48,704	00:00:21,005
3	İŞLEM	5	01.02.2019 15:13:51,666	01.02.2019 15:14:06,688	00:00:15,022
4	YAPANA	6	01.02.2019 15:14:12,685	01.02.2019 15:14:30,668	00:00:17,983
5	YETKİ	5	01.02.2019 15:14:33,665	01.02.2019 15:14:57,000	00:00:23,335
6	BELGESİ	7	01.02.2019 15:14:57,691	01.02.2019 15:15:42,683	00:00:44,992
7	ZORUNLU	7	01.02.2019 15:15:45,715	01.02.2019 15:16:42,706	00:00:56,991
8	OLDU	4	01.02.2019 15:17:09,716	01.02.2019 15:17:21,701	00:00:11,985
				TOPLAM	00:03:11,313
ZUHAL					
1	AD	HARF	BAŞLAMA SÜRESİ	BİTİŞ SÜRESİ	OKUNMA SÜRESİ
2	BORSADA	7	01.02.2019 16:04:37,341	01.02.2019 16:04:55,292	00:00:17,951
3	İŞLEM	5	01.02.2019 16:05:04,291	01.02.2019 16:05:34,306	00:00:30,015
4	YAPANA	6	01.02.2019 16:05:37,292	01.02.2019 16:06:01,308	00:00:24,016
5	YETKİ	5	01.02.2019 16:06:04,304	01.02.2019 16:06:43,308	00:00:39,004
6	BELGESİ	7	01.02.2019 16:06:49,300	01.02.2019 16:07:16,323	00:00:27,023
7	ZORUNLU	7	01.02.2019 16:07:22,313	01.02.2019 16:08:01,328	00:00:39,015
8	OLDU	4	01.02.2019 16:08:10,308	01.02.2019 16:08:19,311	00:00:09,003
				TOPLAM	00:03:06,027
BURAK					
1	AD	HARF	BAŞLAMA SÜRESİ	BİTİŞ SÜRESİ	OKUNMA SÜRESİ
2	BORSADA	7	01.02.2019 16:14:57,720	01.02.2019 16:15:15,698	00:00:17,978
3	İŞLEM	5	01.02.2019 16:15:18,698	01.02.2019 16:15:30,697	00:00:11,999
4	YAPANA	6	01.02.2019 16:15:33,698	01.02.2019 16:16:09,697	00:00:35,999
5	YETKİ	5	01.02.2019 16:16:12,699	01.02.2019 16:16:36,717	00:00:24,018
6	BELGESİ	7	01.02.2019 16:16:39,713	01.02.2019 16:17:09,713	00:00:30,000
7	ZORUNLU	7	01.02.2019 16:17:12,717	01.02.2019 16:17:51,712	00:00:38,995
8	OLDU	4	01.02.2019 16:17:54,715	01.02.2019 16:18:06,697	00:00:11,982
				TOPLAM	00:02:50,971

Literatürde BBA alanında gerçekleştirilmiş olan çalışmaları incelediğimizde ise şu sonuçlarla karşılaşmaktayız.

BBA sistemlerinde performans artırmaya ve iyileştirmeye yönelik yöntemlerin geliştirilmesini incelemiştir. Bu kapasamda BBA sistemlerinin artı eksi yönleri incelenmiş ve karşılaştırılmıştır. Tez çalışması kapsamında ulaşılan sonuçlar neticesinde deneğe özel kısa eğitimler içeren BBA sistemlerinin önümüzdeki dönemlerde sıkça karşımıza çıkacağı vurgulanmıştır.

Beyin bilgisayar arayüzü (BBA) uygulamalarından olan nöro-prostetik bir cihazı kontrol etmesi için tasarlanan bir sistemin incelendiği ve EEG sinyalleri, hesaplanmış olduğu görülmüştür. Bu kapsamda EEG tabanlı BBA sistem uygulamalarının performansının nasıl arttırılabileceği ve öne çıkarılabileceği ortaya konmuştur.

Savaş [65] benzer şekilde EEG tabanlı bir beyin bilgisayar arayüzü (BBA) yazılımı incelenmesi amacıyla model otomobil kontrol yazılımı olan bir beyin bilgisayar arayüzü yazılımının, elektrotlar tarafından ölçülen EEG sinyalleriyle analizini ortaya koymaktadır. Çalışma sonucunda, beyindeki elektriksel sinyallerin ölçülmesi, değerlendirilmesi ile radyo kontrollü bir arabanın hareketlendirilmesi ve ivmeölçer özelliği ile harekete yön verilmesi başarılı bir şekilde gerçekleştirilmiştir. Bu kapsamda EEG cihazları kullanılarak BBA sistemlerinde başarılı sonuçları alınabileceğini ortaya konmuştur.

Yavuz [37] çalışmasında ise EEG sinyalleri kullanılarak bilgisayar ortamında oluşturulmuş bir nesnenin, alfabenin tersten sayılması, küpün zihinsel olarak döndürülmesi, sağ el hareketini düşünülmesi, sol kol hareketinin düşünülmesi ve matematiksel işlemlerin yapılması ile iki eksen üzerinde, sağ, sol, yukarı ve aşağı yönlerde hareket etmesini sağlayan bir beyin-bilgisayar arayüzü tasarlanmış ve bu beyin-bilgisayar arayüzü aracılığı ile gerçek-zamanda, verili yörüngelerin takibi yapılmıştır. Elde edilen sonuçlar karşılaştırıldığında tez çalışmamızla benzer şekilde zihinsel aktivitelerin ölçülmesi ve veri toplanması amacıyla bir BBA sistemi tasarlandığı görülmektedir.

Literatürdeki benzer çalışmalarla karşılaştırıldığında, bu tezin, elde edilen bulgular ve performans açısından doyurucu ve nitelikli olduğu görülmektedir.

KAYNAKÇA

- [1] Albert, W., “Do Web Users Actually Look At Ads? A Case Study Of Banner Ads And Eye-Tracking Technology”, in Proceedings of the Eleventh Annual Conference of Theusability Professionals’ Association, 2002
- [2] Baddeley, A.D., Hitch, G.J.L (1974). Working Memory, In G.A. Bower (Ed.), The psychology of learning and motivation: advances in research and theory, Vol. 8, 47-89.
- [3] Byrne, M. D., Anderson, J. R., Douglas, S., and Matessa, M., “Eye Tracking the Visual earch of Clickdown Menus” in Proceedings of CHI’99 (Pp. 402-409). NY: ACM Press, 1999
- [4] C. Ehmke and S. Wilson, “Identifying web usability problems from eye-tracking data,” Hci 2007, pp. 119–128, 2007.
- [5] Chennamma, H. R. and Yuan, X., “ A survey on eye-gaze tracking techniques”, Indian Journal of Computer Science and Engineering (IJCSE), 4(5): 388-393 (2013).
- [6] Cogain, “ D2.1 Survey of de-facto standards in eye tracking”, Network of Excellence Information Society Technologies,(2005).
- [7] Cornsweet and Crane, “Accurate Two-Dimensional Eye Tracker Using First And Fourth Purkinje Images” in Journal Of The Optical Society Of America. 63:921-928, 1973
- [8] D. Akıncı and K. Çağiltay, “E-devlet Web Sitelerini Kullanmak ya da Kullanamamak : Vatandaş Açısından Kullanılabilirlik Sorunları ve Öneriler,” TBD 21. Ulus. Bilişim Kurult., 2004
- [9] D. Castilla et al., “Effect of Web navigation style in elderly users,” Comput. Human Behav., vol. 55, pp. 909–920, 2016.
- [10] Dodge and Cline, “The Angle Velocity Of Eye Movements”, in Psychological Review, No. 8, pp. 145-157, 1901
- [11] Duchowski and Vertegaal, “Eye-Based Interaction in Graphical Systems: Theory And Practice”, in SIGGRAPH 2000 Course Notes #5. ACM SIGGRAPH, 2000

- [12] Duchowski, A. T. (2003). *Eye Tracking Methodology: Theory and Practice*. London: Springer Verlag.
- [13] Duchowski, A. T. , “ Eye tracking methodology: theory and practice second edition”, Spriger, Clemson, (2006).
- [14] Duchowski, A. T., “ A breadth-first survey of eye tracking applications”, *Behav. Res. Methods Instrum. Comput.*, 34: 1-16 (2002).
- [15] Duchowski, A., “ Eye tracking methodology: theory and practice”, SpringerVerlag, New York, (2003).
- [16] E. Özçelik and K. Çağiltay, “Reklamlarda Bazı Mesajlar Algılanmıyor Mu? Bir Göz Hareketlerini İzleme Çalışması,” in *Ulusal Ergonomi Konferansı*, 2006.
- [17] F. Heidmann, A. Blocher, and M. Burmester, *Usability Engineering*, vol. 1, no. 3/2002. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.
- [18] Fitts, P. M., Jones, R. E. and Milton, J. L., “Eye Movements Of Aircraft Pilots During Instrument-Landing Approaches”, in *Aeronautical Engineering Review*, Vol. 9, Pp. 24-29, 1950
- [19] Goldberg, H. J., & Kotval, X. P. (1999). Computer interface evaluation using eye movements: Methods and constructs. *International Journal of Industrial Ergonomics*, 24, 631-645.
- [20] Goldberg, H. J., and Kotval, X. P., “Computer Interface Evaluation Using Eye Movements: Methods And Constructs”, in *International Journal of Industrial Ergonomics*, 24, 631-645, 1999
- [21] Goldberg, J. H., Stimson, M. J., Lewenstein, M., Scott, N., and Wichansky, A. M., “Eye Tracking in Web Search Tasks: Design Implications”, in *Proceedings of The Eye Tracking Research and Applications Symposium 2002* (pp. 51-58). NY: ACM Press, 2002
- [22] Hansen, D. and Pece, A., “ Eye tracking in the wild”, *Computer Vision and Image Understanding* 98, 1: 155-181 (2005).
- [23] Hansen, D. W. and Ji, Q., “ In the eye of the beholder: A survey of models for eyes and gaze ”, *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(3): 478-500 (2010).

- [24] Hauland, G., “Measuring Team Situation Awareness By Means Of Eye Movement Data”, in Proceedings Of HCI International 2003: Vol 3 (Pp. 230- 234). Mahwah, NJ: Lawrence Erlbaum Associates, 2003
- [25] Huchuan, L., Shipeng, L. and Gang, Y. , “ Robust eye tracking in video sequence”, Journal of Circuits, Systems, and Computers, 21(1): (2012).
- [26] Hyönä, J., Radach, R. & Deubel, H. (2003), The mind's eye: Cognitive and applied aspects of eye movement research , 573-605.
- [27] Iannizzotto, L. and La Rosa, F., “ Competitive combination of multiple eye detection and tracking techniques”, IEEE Transactions on Industrial Electronics, 58(8): 3151-3159 (2011).
- [28] Imai,T.,Sekine,K.,Hattori, K.,Takeda, N.,Koizuka, I.,Nakamae, K.,Miura, K.,Fujioka,H. and Kuboa,T., “ Comparing the accuracy of video-oculography and the scleral search coil system in human eye movement analysis”, Auris Nasus Larynx,32(1): 3-9 (2005).
- [29] J. H. Goldberg, M. J. Stimson, M. Lewenstein, N. Scott, and A. M. Wichansky,(2002) “Eye tracking in web search tasks,” in Proceedings of the symposium on Eye tracking research & applications - ETRA '02, p. 51.
- [30] J. Kim, P. Thomas, and H. Yoon, “Eye-Tracking Analysis of User Behavior and Performance in Web Search on Large and Small Screens,” Commun. Inf. Lit., vol. 3, no. 2, pp. 80–90, 2009.
- [31] J. S. Babcock, M. Lipps, and J. B. Pelz, “How people look at pictures before, during, and after scene capture: Buswell revisited,” Proc. SPIE 4662, pp. 34–47, 2002.
- [32] Jacob, R. K. (1995). Eye tracking in advanced interface design, Virtual environments and advanced interface design, Oxford University Press, Inc., New York, NY.
- [33] John Senders and J.W. Senders , “Visual Scanning Processes”, Book By 1980
- [34] Judd, C.H., Mcallister, C.N. and Steel, W.M., “General Introduction To A Series Of Studies Of Eye Movements By Means Of Kinetoscopic Photographs”, in Psychological Review, Monograph Supplements, No.7, pp. 1-16, 1905

- [35] Just, M. A., & Carpenter, P. A. (1976). Eye fixations and cognitive processes. *Cognitive Psychology*, 8, 441-480.
- [36] K. Majrashi, M. Hamilton, and A. L. Uitdenbogerd, "Correlating Cross-Platform Usability Problems with Eye Tracking Patterns," *Proc. 30th Int. BCS Hum. Comput. Interact. Conf. Fusion!*, no. July, p. 40, 2016.
- [37] K. O. Erdemir and Ö. Yavuz, "Nöro Pazarlamaya Giriş," İstanbul: Brand Map, 2016, pp. 101–106.
- [38] Kenyon, V. R., "A soft contact lens search coil for measuring eye movements", *Vision Research*, 25(11): 1629-1633 (1985).
- [39] Kürşat Çağıltay, *İnsan Bilgisayar Etkileşimi ve Kullanılabilirlik Mühendisliği: Teoriden Pratiğe*. ANKARA: ODTÜ, 2011.
- [40] L. Cowen, L. J. Ball, and J. Delin, "An Eye Movement Analysis of Webpage Usability.," *Proc. HCI 2002*, pp. 317–335, 2002.
- [41] Lupu, R. G. and Ungureanu, F., "A survey of eye tracking methods and applications", *The Polytechnic Institute of Science Bulletin*, (3): 71-86 (2013).
- [42] Lupu, R. G., Bozomitu, R. G., Cehan, V., and Cehan, A., "A new computerbased technology for communicating with people with major neuro-locomotor disability using ocular electromyogram", *34th International Spring Seminar on Electronics Technology, High Tatras*, (2011).
- [43] M. Donmez and K. Cagiltay, "Eye Tracker Based Eye Training System for Children with Low Vision: Effectiveness and Usability," *AERA 2015 Annu. Meet.*, 2015
- [44] Mautone, P. D. & Mayer, R. E. (2001). Signaling as a Cognitive Guide in Multimedia Learning. *Journal of Educational Psychology*, 93, 377-389.
- [45] Mazo, L., Barea, M., Boquete, R., and Lopez, E., "System for assisted mobility using eye movements based on electrooculography", *IEEE Trans.on Neural Sys. and Rehab. Eng.*, 10(4): 209-218 (2002).
- [46] Miles A. Tinker, "Bases For Effective Reading", in *University Of Minnesota Press*, 1965

- [47] Morimoto, C. H. and Mimica, M. R. M., “ Eye gaze tracking techniques for interactive application”, *Computer Vision and Image Understanding* 98, 4-24 (2005).
- [48] Morimoto, C. H., Koons , D., Amir, A., and Flickner, M., “ Pupil detection and tracking using multiple light sources”, *Image and Vision Computing*, 18(4): 331–335 (2000).
- [49] O. Kubus and K. Çağiltay, “E-Devlet Siteleri Görme Engelliler için Erişilebilir mi?,” in *TBD Bilişim Kurultayı*, 2006.
- [50] Oyekoya, O., “Eye Tracking: A Perceptual Interface For Content Based Image Retrieval”, *Phd Thesis, University Of London*, 2007
- [51] Özçelik E., Çağiltay K. (2006). Reklamlarda Bazı Mesajlar Algılanıyor mu? Bir Göz Hareketlerini İzleme Çalışması. Available from
- [52] P. Kasproski and K. Harezlak, “VisDaT – Vision Diagnostics and Treatment System for Children with Disabilities,” 2017.
- [53] Poole, A., Ball, L. J., & Phillips, P. (2004). In search of salience: A response time and eye movement analysis of bookmark recognition. In S. Fincher, P. Markopolous, D.
- [54] Poole, A., Ball, L. J., and Phillips, P., “In Search of Salience: A Response Time And Eye Movement Analysis Of Bookmark Recognition”, in S. Fincher, P. Markopolous, D. Moore, and R. Ruddle (Eds.), *People And Computers XVIII- Design For Life: Proceedings Of HCI 2004*. London: Springer-Verlag Ltd, 2004
- [55] Preston, W. K., Jr., Hutchinson , T. E., and Carley , J. M., “ Spatially dynamic calibration of an eye-tracking system”, *IEEE Transactions on Systems Man and Cybernetics*, 23(4): 1162–1168 (1993).
- [56] R.J.K. Jacob and K.S. Karn, “Eye Tracking in Human-Computer Interaction And Usability Research: Ready To Deliver The Promises (Section Commentary)”, in *The Mind's Eye: Cognitive And Applied Aspects Of Eye Movement Research*, Amsterdam: Elsevier Science, pp. 573-605, 2003
- [57] Rayner K. (1998). Eye Movements in Reading and Information Processing: 20 years of Research, *Psychological Bulletin*, 124, 372-422.

- [58] Redline, C. D., and Lankford, C. P., “Eye-Movement Analysis: A New Tool For Evaluating The Design Of Visually Administered Instruments (Paper And Web)”, in Proceedings of the Section on Survey Research Methods of the American Statistical Association, 2001
- [59] Reulen, J., Marcus, J. T., Kooops, D., de Vries, F. R., Tiesinga, G., Boshuizen, K. and Bos, J.E., “Precise recording of eye movement: the iris technique”, *Med. Biol. Eng. Comput.*, 26(1): 20-26 (1988).
- [60] Ryan, W., “Limbu-track: stable eye-tracking in imperfect light conditions”, Master Thesis, Clemson University, Clemson, (2007).
- [61] S. Djamasbi, M. Siegel, and T. Tullis, “Generation Y, web design, and eye tracking,” *Int. J. Hum. Comput. Stud.*, vol. 68, no. 5, pp. 307–323, 2009.
- [62] S. Djamasbi, T. Tullis, J. Hsu, E. Mazuera, K. Osberg, and J. Bosch, “Gender preferences in web design: usability testing through eye tracking,” *Proc. 13th Am. Conf. Inf. Syst. Keystone Color.*, pp. 1–8, 2007.
- [63] S. Yen and E. Esgin, “Usability Evaluation of Web Based Educational Multimedia by Eye-Tracking Technique,” vol. 5, no. 4, pp. 590–603, 2015.
- [64] Salah, A. A. ve Saygın, A. P. (2008). İnsan ve Bilgisayarda Retinotopi ve Görsel Seçici Dikkat. *Signal Processing, Communication and Applications Conference, 2008. SIU 2008. IEEE 16th.* 1-6.
- [65] Savaş, Z., “Real-time detection and tracking of human eyes in video sequences”, Doktora Tezi, Ortadoğu Teknik Üniversitesi, Ankara, (2005).
- [66] Savaş, Z., “Real-time detection and tracking of human eyes in video sequences”, Doktora Tezi, Ortadoğu Teknik Üniversitesi, Ankara, (2005).
- [67] Schnell, T., Wu, T., “Applying Eye Tracking As Alternative Approach For Activation Of Controls And Functions in Aircraft”, in Proceedings of The 5th International Conference on Human Interaction with Complex Systems (HICS), April, 30 - May, 2, 2000, Urbana, Illinois, USA, pp 113, 2000
- [68] Scott, D. and Findlay, J. M., “Visual search, eye movements and display units”, *Human factors report.*, South Road, (1993).

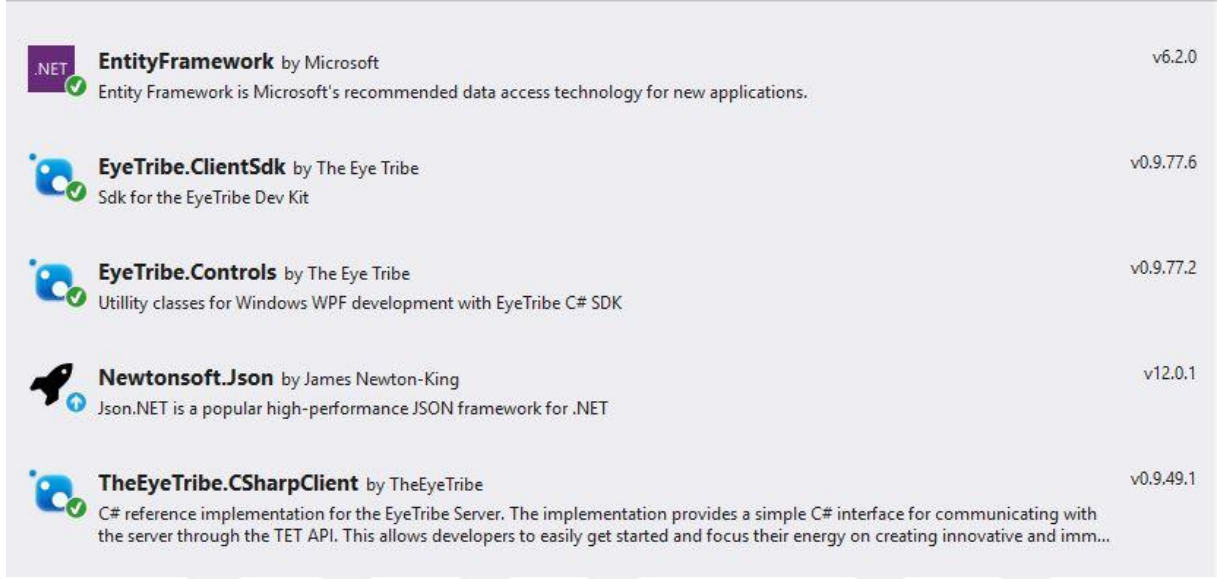
- [69] Septanto, H., Prihatmanto, A. S. , and Indrayanto, A., “ A computer cursor controlled by eye movements and voluntary eye winks using a single channel EOG”, International Conference on Electrical Engineering and Informatics, Bangi Selangor, (2009).
- [70] Sigut, J. and Sidha, S. A., “ Iris center corneal reflection method for gaze tracking using visible light”, IEEE Transactions On Biomedical Engineering, 58(2): 411-419 (2011)
- [71] Snodderly, D. M., Kagan, I., and Gur, M., “Selective Activation Of Visual Cortex Neurons By Fixational Eye Movements: Implications For Neural Coding”, in Visual Neuroscience, Vol.18, pp. 259-277, 2001
- [72] Sohel Merchant, (2001) Eye Movement Research in Aviation And Commercially Available Eye Trackers Today
- [73] Solso, R. L., Maclin, M. K., Maclin, O. H. (2007). Bilişsel Psikoloji. Kitabevi Yayınları.
- [74] Tan, K. H., Kriegman, D. J., and Ahuja, N., “ Appearance-based eye gaze estimation”, Applications of Computer Vision (WACV 2002).Proceedings, (2002).
- [75] Torricelli, D., Conforto, S., and Schmid, M., “ A neural-based remote eye gaze tracker under natural head motion”, Computer Methods and Programs in Biomedicine, 9(2): 66-78 (2008).
- [76] Tsotsos, J.K., Culhane, S.M., Wai, W.Y.K., Lai, Y.H., Davis, N.ve Nuflo, F. (1995). Modelling Visual Attention via Selective Tuning, Artificial Intelligence, Vol. 78, 507–545.
- [77] Vertegaal, R., “The GAZE Groupware System: Mediating Joint Attention in Mutiparty Communication And Collaboration” , in Human Factors in Computing Systems: CHI '99 Conference Proceedings (P. 294-301). ACM Press, 1999
- [78] Wild, D. J., “ Gaze tracking using a regular web camera”, Bachelor of Science, Rhodes University, Grahamstown, (2012).
- [79] Wyszecki, G. and Stiles, W. , “ Color science: concepts and methods, quantitative data and formulae” , John Wiley & Sons, New York, (1992).
- [80] Web site <http://theeyetribe.com> cihaz görselleri için alınan kaynak.

- [81] Yamada, M. and Fukuda, T., “ Eye word processor (EWP) and peripheral controller for the ALS patient”, Physical Science, Measurement and Instrumentation-Reviews IEEE Proceedings A, 134: 328-330 (1987).
- [82] Yarbus, A. L., “ Eye movements and vision” , Plenum Press, New York, (1967).
- [83] Yıldız, F.ve Baykan, N. A., “ Çapraz ilişki metoduyla iris tanıma”, Journal of Technical-Online Selçuk Üniversitesi, 10(1): 19-37 (2011).
- [84] Yıldız, M., “ A new coding technique for EOG based writing systems”, 19th Signal Processing and Communications Applications Conference (SIU), Antalya, (2011).
- [85] Yurdugül, H & Aşkar, P. (2004). Ortaöğretim Kurumları Öğrenci Seçme ve Yerleştirme Sınavı'nın cinsiyete göre madde yanlılığı açısından incelenmesi. Eğitim Bilimleri ve Uygulama Dergisi, 3(5), 3-20.
- [86] Yunus Çobas, Adil Deniz Duru, Osman Nuri Uçan, Oğuz Bayat (2017) Dikkat seviyesinin gerçek zamanda izlenmesi (ALTINBAŞ ÜNİVERSİTESİ MÜHENDİSLİK SİSTEMLERİ VE MİMARLIK DERGİSİ)
- [87] Zhu, Z. and Ji , Q., “ Novel eye gaze tracking techniques under natural head movement”, IEEE Transactions on Biomedical Engineering, 54(12): 2246- 2260 (2007).
- [88] Zhu, Z. and Ji, Q., “ Eye and gaze tracking for interactive graphic display”, Machine Vision and Applications, 15(3): 139-148 (2002).
- [89] Zacheboatrental.com web sitesi görseli

EK A.

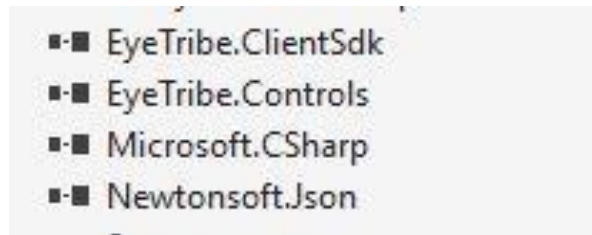
EYE TRIBE SDK (SOFTWARE DEVELOPMENT KIT) YÜKLENMESİ

C# dili programlama için kullandığımız Microsoft Visual Studio ortamına Eye Tribe SDK (software development kit) eklenmelidir. Microsoft Visual Studio, Microsoft tarafından geliştirilen ve bir çok yazılım dilini destekleyen bir derleyici programıdır.



Şekil A.1 : SDK (Software Development Kit) Eklenmesi

Visual Studio ortamında kod geliştirdiğimiz için ekranında sağ tarafta bulunan “Solution” içerik ağacının “Reference” sekmesini sağ tıklayarak referans ekle seçeneğini seçiyoruz.



Şekil A.2 : Referans Edilmiş SDK Projeye Dahil Etmek

EK B.

5.2. UYGULAMA KOD ÇIKTISI

```
private void button1_Click(object sender, EventArgs e)
{
    bool baglanti = Connect("localhost", 6555);
    timer1.Enabled = true;
    timer1.Start();
    Send("get");
    ListenerLoop();
}

private void Send(string message)
{
    if (socket != null && socket.Connected)
    {
        StreamWriter writer = new StreamWriter(socket.GetStream());
        writer.WriteLine(message);
        writer.Flush();
    }
}

private async void ListenerLoop()
{
    StreamReader reader = new StreamReader(socket.GetStream());
    isRunning = true;

    while (isRunning)
    {
        string response = string.Empty;

        try
        {
            response = await reader.ReadLineAsync();

            File file = new File();
            byte[] i = new byte[0];
        }
    }
}
```

```

StringBuilder sbLog = new StringBuilder();
sbLog.AppendLine(response);
file.Write("C:\\Loglama\\", "Log.txt", sbLog.ToString());

textBox1.Text = response.ToString() + Environment.NewLine;

foreach (DataRow dr in jDt.JsonStringToDataTable(response).Rows)
{
    if (dr.ItemArray.Length == 19)
    {
        v = new veriler();
        if (dr["category"].ToString() != "heartbeat")
        {
            v.Category = dr["category"].ToString();
            if (dr["request"].ToString() != null)
                v.ReguestSorgu = dr["request"].ToString();

            if (Convert.ToInt32(dr["statuscode"].ToString()) > 0)
                v.statusCode = Convert.ToInt32(dr["statuscode"].ToString());

            if (dr["values"].ToString() != null)
                v.ValuesParametre = dr["values"].ToString();

            if (dr["fix"].ToString() != null)
                v.Fix = Convert.ToBoolean(dr["fix"]);

            if (dr["lefteye"].ToString() != null)
                v.Lefteye = dr["lefteye"].ToString();
            else
                v.Lefteye = string.Empty;

            if (dr["lefteye y"].ToString() != null)
                v.LefteyeY = dr["lefteye y"].ToString();
            else
                v.LefteyeY = string.Empty;
        }
    }
}

```

```

if (dr["pcenter"].ToString() != null)
    v.pCenter = dr["pcenter"].ToString();

if (dr["psize"].ToString() != null)
    v.psize = Convert.ToDecimal(dr["psize"].ToString());
else
    v.psize = 0;
if (dr["raw"].ToString() != null)
    v.raw = dr["raw"].ToString();

if (dr["righteye"].ToString() != null)
    v.righteye = dr["righteye"].ToString();
else
    v.righteye = string.Empty;

if (dr["righteye y"].ToString() != null)
    v.righteyeY = dr["righteye y"].ToString();
else
    v.righteyeY = string.Empty;

if (dr["state"].ToString() != null)
    v.state = Convert.ToInt32(dr["state"].ToString());

if (dr["time"].ToString() != null)
    v.time = Convert.ToDouble(dr["time"].ToString());
else
    v.state = 0;
if (dr["timestamp"].ToString() != null)
    v.timestamp = Convert.ToDateTime(dr["timestamp"].ToString());

eklenecek.Add(v);
}
}
}

```

```

    }
    catch (Exception ex)
    {
        Console.Out.WriteLine("Error while reading response: " + ex.Message);
    }
    finally
    {
        if (timer1.Enabled == false)
            socket.Close();
    }
}
}

```

```
private async void timer1_Tick(object sender, EventArgs e)
```

```

{
    dataGridView1.DataSource = await GetListAsync();
    label2.Text = eklenecek.Count.ToString();
}

```

```
private Task<List<veriler>> GetListAsync()
```

```

{
    return Task.Run(() => eklenecek.ToList());
}

```

```
private void button2_Click(object sender, EventArgs e)
```

```

{
    timer1.Stop();
    timer1.Enabled = false;

    timer1.Stop();
    timer1.Enabled = false;
    Application.Exit();
}

```


EK C.

5.3 UYGULAMA KOD ÇIKTISI

```
public bool Connect(string host, int port)
{
    try
    {
        socket = new TcpClient("localhost", 6555);
    }
    catch (Exception ex)
    {
        Console.Out.WriteLine("Error connecting: " + ex.Message);
        return false;
    }
    string REQ_CONNECT =
    "{\"values\":{\"push\":true,\"version\":1},\"category\":\"tracker\",\"request\":\"set\"}";
    Send(REQ_CONNECT);

    incomingThread = new Thread(ListenerLoop);
    incomingThread.Start();
    string REQ_HEATBEAT = "{\"category\":\"heartbeat\",\"request\":\"get\"}";
    timerHeartbeat = new System.Timers.Timer(250);
    timerHeartbeat.Elapsed += delegate { Send(REQ_HEATBEAT); };
    timerHeartbeat.Start();

    return true;
}

private async void ListenerLoop()
{
    StreamReader reader = new StreamReader(socket.GetStream());
    isRunning = true;
    decimal iX;
    decimal iY;
    byte iIntense;
    Bitmap bMap = new Bitmap(pictureBox1.Width, pictureBox1.Height);
```

```

Random rRand = new Random();

while (isRunning)
{
    string response = string.Empty;

    try
    {
        response = await reader.ReadLineAsync();

        foreach (DataRow dr in jDt.JsonStringToDataTable(response.ToString()).Rows)
        {
            v = new veriler();

            if (dr["category"].ToString() != "heartbeat")
            {
                iX = Convert.ToDecimal(dr["lefteye"].ToString().Replace("avg:x:", "").Replace(".",
                ","));
                iY = Convert.ToDecimal(dr["lefteye y"].ToString().Replace(".", ","));
                iIntense = (byte)rRand.Next(0, 254);
                HeatPoints.Add(new HeatPoint(iX, iY, iIntense));
                iX = Convert.ToDecimal(dr["righteye"].ToString().Replace("avg:x:", "").Replace(".",
                ","));
                iY = Convert.ToDecimal(dr["righteye y"].ToString().Replace(".", ","));
                iIntense = (byte)rRand.Next(0, 254);
                HeatPoints.Add(new HeatPoint(iX, iY, iIntense));
            }
        }
        pictureBox1.Image = await CreateIntensityMask(bMap, HeatPoints);
    }
    catch (Exception ex)
    {
        Console.Out.WriteLine("Error while reading response: " + ex.Message);
    }
}

```

```
}
```

```
private void Send(string message)
```

```
{
```

```
if (socket != null && socket.Connected)
```

```
{
```

```
StreamWriter writer = new StreamWriter(socket.GetStream());
```

```
writer.WriteLine(message);
```

```
writer.Flush();
```

```
}
```

```
}
```

```
private async Task<Bitmap>CreateIntensityMask(Bitmap bSurface, List<HeatPoint>  
aHeatPoints)
```

```
{
```

```
Graphics DrawSurface = Graphics.FromImage(bSurface);
```

```
DrawSurface.Clear(Color.White);
```

```
foreach (HeatPoint DataPoint in aHeatPoints)
```

```
{
```

```
DrawHeatPoint(DrawSurface, DataPoint, 15);
```

```
}
```

```
return bSurface;
```

```
}
```

```
private void DrawHeatPoint(Graphics Canvas, HeatPoint HeatPoint, int Radius)
```

```
{
```

```
List<Point> CircumferencePointsList = new List<Point>();
```

```
Point CircumferencePoint;
```

```
Point[] CircumferencePointsArray;
```

```
float fRatio = 1F / Byte.MaxValue;
```

```
byte bHalf = Byte.MaxValue / 2;
```

```
int iIntensity = (byte)(HeatPoint.Intensity - ((HeatPoint.Intensity - bHalf) * 2));
```

```
float fIntensity = iIntensity * fRatio;
```

```
for (double i = 0; i <= 360; i += 10)
```

```

{

    CircumferencePoint = new Point();
    CircumferencePoint.X = Convert.ToInt32((Convert.ToInt32(HeatPoint.X) + Radius *
Math.Cos(ConvertDegreesToRadians(i))));
    CircumferencePoint.Y = Convert.ToInt32(Convert.ToInt32(HeatPoint.Y) + Radius *
Math.Sin(ConvertDegreesToRadians(i)));
    CircumferencePointsList.Add(CircumferencePoint);
}
CircumferencePointsArray = CircumferencePointsList.ToArray();
PathGradientBrush GradientShaper = new PathGradientBrush(CircumferencePointsArray);
ColorBlend GradientSpecifications = new ColorBlend(3);
GradientSpecifications.Positions = new float[3] { 0, fIntensity, 1 };
GradientSpecifications.Colors = new Color[3]
{
    Color.FromArgb(0, Color.White),
    Color.FromArgb(HeatPoint.Intensity, Color.Black),
    Color.FromArgb(HeatPoint.Intensity, Color.Black)
};
GradientShaper.InterpolationColors = GradientSpecifications;
Canvas.FillPolygon(GradientShaper, CircumferencePointsArray);
}

private double ConvertDegreesToRadians(double degrees)
{
    double radians = (Math.PI / 180) * degrees;
    return (radians);
}

```

EK D.

5.4 UYGULAMA KOD ÇIKTISI

```
private void button9_Click_1(object sender, EventArgs e)
{
    MessageBox.Show("Lütfen 1 Bakınız");
    Calismak();
}

public void Calismak()
{
    bool baglanti = Connect("localhost", 6555);
    timer1.Enabled = true;
    timer1.Start();
    ListenerLoop();
}

string JsonString = string.Empty;
ConvertJsonStringToDataTable jDt = new ConvertJsonStringToDataTable();
veriler v;
List<veriler> eklenecek = new List<veriler>();
private List<HeatPointButtonForm> HeatPoints = new List<HeatPointButtonForm>();
private List<HeatPointButtonForm> HeatPointsEkle = new
List<HeatPointButtonForm>();
DialogResult okm;

private async void ListenerLoop()
{
    StreamReader reader = new StreamReader(socket.GetStream());
    isRunning = true;
    int count = 0;
    Double iX;
    Double iY;
```

```

while (isRunning)
{
    string response = string.Empty;

    try
    {
        response = await reader.ReadLineAsync();

        textBox1.Text = response.ToString() + Environment.NewLine;

        File file = new File();
        byte[] i = new byte[0];
        StringBuilder sbLog = new StringBuilder();
        sbLog.AppendLine(response);

        file.Write("C:\\Loglama\\", "Log.txt", sbLog.ToString());
        foreach (DataRow dr in jDt.JsonStringToDataTable(response).Rows)
        {
            if (dr.ItemArray.Length == 19)
            {
                v = new veriler();
                if (dr["category"].ToString() != "heartbeat")
                {
                    iX    = Convert.ToDouble(dr["lefteye"].ToString().Replace("avg:x:",
"".Replace(".", ",")));
                    iY = Convert.ToDouble(dr["lefteye y"].ToString().Replace(".", ","));

                    HeatPoints.Add(new HeatPointButtonForm(iX, iY));

                    iX    = Convert.ToDouble(dr["righteye"].ToString().Replace("avg:x:",
"".Replace(".", ",")));
                    iY = Convert.ToDouble(dr["righteye y"].ToString().Replace(".", ","));
                    HeatPoints.Add(new HeatPointButtonForm(iX, iY));
                }
            }
        }
    }
}

```

```

if (((iX >= 509.0000) && (iX <= 545.9999)) || ((iY >= 509.0000) && (iY
<= 545.9999)))
{
    count = count + 1;
    Thread.Sleep(2000);
    if (count == 1)
    {
        okm=  MessageBox.Show("button 1 baktınız", "Bildirim",
MessageBoxButtons.OKCancel, MessageBoxIcon.Information);
        if (okm.ToString() == "OK")
        {
            string mesajbirinci = "button 1 baktınız";
            if (count <2)
            {
                KayitEkleSoru("Buton 1 Bakınız",
DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"));
                KayitEkle(mesajbirinci, DateTime.UtcNow.ToString("yyyy-
MM-dd HH:mm:ss.fff"), "Buton 1 Bakınız");
                count = 2;
                isRunning = false;
                okm =  MessageBox.Show("button 2 Bakınız", "Bildirim",
MessageBoxButtons.OKCancel, MessageBoxIcon.Information);
                KayitEkleSoru("Buton 2 Bakınız",
DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"));
                if (okm.ToString() == "OK")
                {
                    ListenerLoopIki();
                    break;
                }
            }
            break;
        }
        break;
    }
}

```

```

        }
    }
}
}
break;
}
}
catch (Exception ex)
{
    Console.Out.WriteLine("Error while reading response: " + ex.Message);
}
finally
{
    if (timer1.Enabled == false)
        socket.Close();
}
}
}

private async void ListenerLoopIki()
{
    StreamReader reader = new StreamReader(socket.GetStream());
    isRunning = true;
    int count = 0;
    Double iX;
    Double iY;
    while (isRunning)
    {
        string response = string.Empty;

        try
        {
            response = await reader.ReadLineAsync();

```



```

textBox1.Text = response.ToString() + Environment.NewLine;

File file = new File();
byte[] i = new byte[0];
StringBuilder sbLog = new StringBuilder();
sbLog.AppendLine(response);

file.Write("C:\\Loglama\\", "Log2.txt", sbLog.ToString());
foreach (DataRow dr in jDt.JsonStringToDataTable(response).Rows)
{
    if (dr.ItemArray.Length == 19)
    {
        v = new veriler();
        if (dr["category"].ToString() != "heartbeat")
        {
            iX = Convert.ToDouble(dr["lefteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
            iY = Convert.ToDouble(dr["lefteye y"].ToString().Replace(".", ","));

            HeatPoints.Add(new HeatPointButtonForm(iX, iY));
            iX = Convert.ToDouble(dr["righteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
            iY = Convert.ToDouble(dr["righteye y"].ToString().Replace(".", ","));
            HeatPoints.Add(new HeatPointButtonForm(iX, iY));
            if (((iX >= 1189.0000) && (iX <= 1243.9999)) || ((iY >= 1170.0000) &&
(iY <= 1246.9999)))
            {
                count = count + 1;
                Thread.Sleep(2000);
                if (count == 1)
                {

```



```

catch (Exception ex)
{
    Console.Out.WriteLine("Error while reading response: " + ex.Message);
}
finally
{
    if (timer1.Enabled == false)
        socket.Close();
}
}
}

private async void ListenerLoopUC()
{
    StreamReader reader = new StreamReader(socket.GetStream());
    isRunning = true;
    int count = 0;
    Double iX;
    Double iY;
    while (isRunning)
    {
        string response = string.Empty;

        try
        {
            response = await reader.ReadLineAsync();

            textBox1.Text = response.ToString() + Environment.NewLine;

            File file = new File();
            byte[] i = new byte[0];
            StringBuilder sbLog = new StringBuilder();

```

```

sbLog.AppendLine(response);

file.Write("C:\\Loglama\\", "Log.txt", sbLog.ToString());
foreach (DataRow dr in jDt.JsonStringToDataTable(response).Rows)
{
    if (dr.ItemArray.Length == 19)
    {
        v = new veriler();
        if (dr["category"].ToString() != "heartbeat")
        {
            iX = Convert.ToDouble(dr["lefteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
            iY = Convert.ToDouble(dr["lefteye y"].ToString().Replace(".", ","));

            HeatPoints.Add(new HeatPointButtonForm(iX, iY));
            iX = Convert.ToDouble(dr["righteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
            iY = Convert.ToDouble(dr["righteye y"].ToString().Replace(".", ","));
            HeatPoints.Add(new HeatPointButtonForm(iX, iY));
            if (((iX >= 179.0000) && (iX <= 182.9999)) || ((iY >= 152.0000) && (iY
<= 155.9999)))
            {
                count = count + 1;
                Thread.Sleep(2000);
                if (count == 1)
                {
                    okm = MessageBox.Show("button 3 baktınız", "Bildirim",
MessageBoxButtons.OKCancel, MessageBoxIcon.Information);
                    if (okm.ToString() == "OK")
                    {
                        string mesajbirinci = "button 3 baktınız";
                        if (count < 2)
                        {

```



```

        if (timer1.Enabled == false)
            socket.Close();
    }
}
}
public void KayitEkle(string mesaj, string zaman, string sorular)
{
    SqlConnection baglan = new SqlConnection("Data Source=DESKTOP-
FG7D15E\\SQLEXPRESS;Initial Catalog=OkulTez;Integrated Security=True");
    SqlCommand _cmd = new SqlCommand("insert into hipotez
(buttonmesaj,buttonsuresi,soru) values('" + mesaj + "','" + zaman + "','" + sorular + "')", baglan);
    try
    {
        baglan.Open();
        _cmd.ExecuteNonQuery();
        baglan.Close();
        baglan.Dispose();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
    }
}
public void KayitEkleSoru(string soru, string zaman)
{
    SqlConnection baglan = new SqlConnection("Data Source=DESKTOP-
FG7D15E\\SQLEXPRESS;Initial Catalog=OkulTez;Integrated Security=True");
    SqlCommand _cmd = new SqlCommand("insert into Sorular (soru,tarih) values('" +
soru + "','" + zaman + "')", baglan);
    try
    {
        baglan.Open();

```

```
        _cmd.ExecuteNonQuery();
        baglan.Close();
        baglan.Dispose();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
    }
}
```

```
int i = 0;
int count = 1;

private TcpClient socket;
private Thread incomingThread;
private System.Timers.Timer timerHeartbeat;
public bool isRunning;
void Yukleme()
{
    timer1.Interval = 2000;
    timer1.Start();
    button1.Width = 60;
    button1.Height = 40;
    button1.Visible = false;
    Point konum = button1.Location;
    konumX = konum.X;
    konumY = konum.Y;
    KonumXX = konum.X + 60;
    KonumYY = konum.Y + 40;

    timer2.Interval = 8000;
    timer2.Start();
}
```

```

        KayitEkleSoru(DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"),
        "BUTON OLUŞTU");
    }
    public double konumX, konumY, KonumXX, KonumYY;

    private void timer2_Tick(object sender, EventArgs e)
    {
        bool baglanti = Connect("localhost", 6555);
        ListenerLoop();
        timer2.Stop();
    }
    private void timer1_Tick(object sender, EventArgs e)
    {
        button1.Visible = true;
        timer1.Stop();
    }
    public bool Connect(string host, int port)
    {
        try
        {
            socket = new TcpClient("localhost", 6555);
        }
        catch (Exception ex)
        {
            Console.Out.WriteLine("Error connecting: " + ex.Message);
            return false;
        }
    }

    string REQ_CONNECT =
    "{\"values\":{\"push\":true,\"version\":1},\"category\":\"tracker\",\"request\":\"set\"}";
    Send(REQ_CONNECT);

    incomingThread = new Thread(ListenerLoop);

```



```

incomingThread.Start();

string REQ_HEATBEAT = "{\"category\":\"heartbeat\",\"request\":\"get\"}";
timerHeartbeat = new System.Timers.Timer(250);
timerHeartbeat.Elapsed += delegate { Send(REQ_HEATBEAT); };
timerHeartbeat.Start();

return true;
}
private void Send(string message)
{
    if (socket != null && socket.Connected)
    {
        StreamWriter writer = new StreamWriter(socket.GetStream());
        writer.WriteLine(message);
        writer.Flush();
    }
}
public event EventHandler<ReceivedDataEventArgs> OnData;
DataTable dt;
string jsonString = string.Empty;
ConvertJsonStringToDataTable jDt = new ConvertJsonStringToDataTable();
veriler v;
List<veriler> eklenecek = new List<veriler>();
private List<HeatPointButtonFormYen> HeatPoints = new
List<HeatPointButtonFormYen>();

public int counter = 0;
private async void ListenerLoop()
{
    StreamReader reader = new StreamReader(socket.GetStream());

```

```

isRunning = true;
decimal iX;
decimal iY;
byte iIntense;

Random rRand = new Random();

while (isRunning)
{
    string response = string.Empty;

    try
    {
        response = await reader.ReadLineAsync();

        // textBox1.Text = response + Environment.NewLine;

        foreach (DataRow dr in jDt.JsonStringToDataTable(response.ToString()).Rows)
        {
            v = new veriler();

            if (dr["category"].ToString() != "heartbeat")
            {

                iX    =    Convert.ToDecimal(dr["lefteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
                iY = Convert.ToDecimal(dr["lefteye y"].ToString().Replace(".", ","));
                iIntense = (byte)rRand.Next(0, 254);
                HeatPoints.Add(new HeatPointButtonFormYen(iX, iY));
                iX    =    Convert.ToDecimal(dr["righteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
                iY = Convert.ToDecimal(dr["righteye y"].ToString().Replace(".", ","));
                iIntense = (byte)rRand.Next(0, 254);
            }
        }
    }
}

```

```

HeatPoints.Add(new HeatPointButtonFormYen(iX, iY));

foreach (var item in HeatPoints)
{
    if ((item.X >= Convert.ToDecimal(konumX) && (item.X <=
Convert.ToDecimal(KonumXX)) && (item.Y >= Convert.ToDecimal(konumY) && (item.Y
<= Convert.ToDecimal(KonumYY))))
    {
        counter = counter + 1;
        if (counter == 1)
        {
            Tiklamak(counter);
            break;
        }
        else
        {
            break;
        }
    }
}
counter = 0;
break;
}
}

catch (Exception ex)
{
    Console.Out.WriteLine("Error while reading response: " + ex.Message);
}
}
}

void Tiklamak(int id)

```

```

{
    // KayitEkleSoru(DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"), "Buton
Oluştur.");
    if ((counter==1)&&(id==1)&&(counter==1))
    {
        Random x = new Random();
        i++;
        Point pt = new Point(
            int.Parse(x.Next(Form1.ActiveForm.Width / 2).ToString()),
            int.Parse(x.Next(Form1.ActiveForm.Height / 2).ToString())
        );
        count = count + 1;
        button1.Location = pt;
        button1.Width = 60;
        button1.Height = 40;
        button1.Text = "Buton ";//+ count.ToString();
        Thread.Sleep(3000);

        konumX = pt.X;
        konumY = pt.Y;
        KonumXX = konumX + 60;
        KonumYY = konumY + 40;
        KayitEkleSoru(DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"), "Buton
Tıklandı");
        String filename = "C:\\Windows\\Media\\tada.wav";
        System.Media.SoundPlayer sp = new System.Media.SoundPlayer(filename);
        sp.PlaySync();
        MessageBox.Show("Buton Tıklandı");
    }
}

private void Button1_Click(object sender, EventArgs e)
{
    Random x = new Random();

```

```

        i++;
        Point pt = new Point(
            int.Parse(x.Next(Form1.ActiveForm.Width / 2).ToString()),
            int.Parse(x.Next(Form1.ActiveForm.Height / 2).ToString())
        );
        count = count + 1;
        button1.Location = pt;
        button1.Width = 60;
        button1.Height = 40;
        button1.Text = "Buton " + count.ToString();
        //KayitEkleSoru(DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"), "Buton
Oluşturdu.");
        KayitEkleSoru(DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"),
"Button Tıklandı");
        String filename = "C:\\Windows\\Media\\tada.wav";
        System.Media.SoundPlayer sp = new System.Media.SoundPlayer(filename);
        sp.PlaySync();
        MessageBox.Show("Buton Tıklandınız");
    }
    public struct HeatPointButtonFormYen
    {
        public decimal X;
        public decimal Y;

        public HeatPointButtonFormYen(decimal iX, decimal iY)
        {
            X = iX;
            Y = iY;
        }
    }
}

```

EK E.

5.5 UYGULAMA KOD ÇIKTISI

```
int i = 0;
int count = 1;
private TcpClient socket;
private Thread incomingThread;
private System.Timers.Timer timerHeartbeat;
public bool isRunning;
void Yukleme()
{
    timer1.Interval = 2000;

    timer1.Start();

    button1.Width = 60;

    button1.Height = 40;

    button1.Visible = false;

    Point konum = button1.Location;

    konumX = konum.X;

    konumY = konum.Y;

    KonumXX = konum.X + 60;

    KonumYY = konum.Y + 40;

    timer2.Interval = 8000;
```

```

timer2.Start();

KayitEkleSoru(DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"), "BUTON
OLUŞTU");
}

public double konumX, konumY, KonumXX, KonumYY;
private void timer2_Tick(object sender, EventArgs e)
{
    bool baglanti = Connect("localhost", 6555);
    ListenerLoop();
    timer2.Stop();
}

private void timer1_Tick(object sender, EventArgs e)
{
    button1.Visible = true;
    timer1.Stop();
}

public bool Connect(string host, int port)
{
    try
    {
        socket = new TcpClient("localhost", 6555);
    }
    catch (Exception ex)
    {
        Console.Out.WriteLine("Error connecting: " + ex.Message);
        return false;
    }
    string REQ_CONNECT =
"{\"values\":{\"push\":true,\"version\":1},\"category\":\"tracker\",\"request\":\"set\"}";
    Send(REQ_CONNECT);
}

```

```

        incomingThread = new Thread(ListenerLoop);
        incomingThread.Start();
        string REQ_HEATBEAT = "{\"category\":\"heartbeat\",\"request\":\"get\"}";
        timerHeartbeat = new System.Timers.Timer(250);
        timerHeartbeat.Elapsed += delegate { Send(REQ_HEATBEAT); };
        timerHeartbeat.Start();
        return true;
    }
    private void Send(string message)
    {
        if (socket != null && socket.Connected)
        {
            StreamWriter writer = new StreamWriter(socket.GetStream());
            writer.WriteLine(message);
            writer.Flush();
        }
    }
    public event EventHandler<ReceivedDataEventArgs> OnData;
    DataTable dt;
    string JsonString = string.Empty;
    ConvertJsonStringToDataTable jDt = new ConvertJsonStringToDataTable();
    veriler v;
    List<veriler> eklenecek = new List<veriler>();
    private List<HeatPointButtonFormYen> HeatPoints = new List<HeatPointButtonFormYen>();
    public int counter = 0;
    private async void ListenerLoop()
    {
        StreamReader reader = new StreamReader(socket.GetStream());
        isRunning = true;
        decimal iX;
        decimal iY;
        byte iIntense;
        Random rRand = new Random();

```



```

while (isRunning)
{
    string response = string.Empty;
    try
    {
        response = await reader.ReadLineAsync();
        foreach (DataRow dr in
jDt.JsonStringToDataTable(response.ToString()).Rows)
        {
            v = new veriler();
            if (dr["category"].ToString() != "heartbeat")
            {
                iX = Convert.ToDecimal(dr["lefteye"].ToString().Replace("avg:x:",
"".Replace(".", ",")));
                iY = Convert.ToDecimal(dr["lefteye y"].ToString().Replace(".", ","));
                iIntense = (byte)rRand.Next(0, 254);
                HeatPoints.Add(new HeatPointButtonFormYen(iX, iY));
                iX = Convert.ToDecimal(dr["righteye"].ToString().Replace("avg:x:",
"".Replace(".", ",")));
                iY = Convert.ToDecimal(dr["righteye y"].ToString().Replace(".", ","));
                iIntense = (byte)rRand.Next(0, 254);
                HeatPoints.Add(new HeatPointButtonFormYen(iX, iY));
                foreach (var item in HeatPoints)
                {
                    if ((item.X >= Convert.ToDecimal(konumX) && (item.X <=
Convert.ToDecimal(KonumXX)) && (item.Y >= Convert.ToDecimal(konumY) && (item.Y
<= Convert.ToDecimal(KonumYY))))
                    {
                        counter = counter + 1;
                        if (counter == 1)
                        {
                            Tiklamak(counter);
                            break;
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    else
    {
        break;
    }
}
counter = 0;
break;
}
}
}
catch (Exception ex)
{
    Console.Out.WriteLine("Error while reading response: " + ex.Message);
}
}
}

```

```
void Tiklamak(int id)
```

```

{
    if ((counter==1)&&(id==1)&&(counter==1))
    {
        Random x = new Random();
        i++;
        Point pt = new Point(
            int.Parse(x.Next(Form1.ActiveForm.Width / 2).ToString()),
            int.Parse(x.Next(Form1.ActiveForm.Height / 2).ToString())
        );
        count = count + 1;
        button1.Location = pt;
        button1.Width = 60;
        button1.Height = 40;
        button1.Text = "Button " + count.ToString();
    }
}

```

```

Thread.Sleep(3000);
konumX = pt.X;
konumY = pt.Y;
KonumXX = konumX + 60;
KonumYY = konumY + 40;
KayitEkleSoru(DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"), "Buton
Tıklandı");
String filename = "C:\\Windows\\Media\\tada.wav";
System.Media.SoundPlayer sp = new System.Media.SoundPlayer(filename);
sp.PlaySync();
MessageBox.Show("Button Tıklandı");
}
}

private void Button1_Click(object sender, EventArgs e)
{
Random x = new Random();
i++;
Point pt = new Point(
int.Parse(x.Next(Form1.ActiveForm.Width / 2).ToString()),
int.Parse(x.Next(Form1.ActiveForm.Height / 2).ToString())
);
count = count + 1;
button1.Location = pt;
button1.Width = 60;
button1.Height = 40;
button1.Text = "Button " + count.ToString();
KayitEkleSoru(DateTime.UtcNow.ToString("yyyy-MM-dd HH:mm:ss.fff"), "Buton
Tıklandı");
String filename = "C:\\Windows\\Media\\tada.wav";
System.Media.SoundPlayer sp = new System.Media.SoundPlayer(filename);
sp.PlaySync();
MessageBox.Show("Buton Tıkladınız");
}

```

```
}
```

```
public struct HeatPointButtonFormYen
```

```
{
```

```
    public decimal X;
```

```
    public decimal Y;
```

```
    public HeatPointButtonFormYen(decimal iX, decimal iY)
```

```
    {
```

```
        X = iX;
```

```
        Y = iY;
```

```
    }
```

```
}
```



EK F.

5.6 UYGULAMA KOD ÇIKTISI

```
public Form8()
```

```
{
```

```
    InitializeComponent();
```

```
    AHarfi();
```

```
    QHarfi();
```

```
    ZHarfi();
```

```
    WHarfi();
```

```
    EHarfi();
```

```
    RHarfi();
```

```
    THarfi();
```

```
    YHarfi();
```

```
    UHarfi();
```

```
    IHarfi();
```

```
    OHarfi();
```

```
    PHarfi();
```

```
    ĞHarfi();
```

```
    ÜHarfi();
```

```
    SHarfi();
```

```
    DHarfi();
```

```
    FHarfi();
```

```
    GHarfi();
```

```
    HHarfi();
```

```
    JHarfi();
```

```
    KHarfi();
```

```
    LHarfi();
```

```
    ŞHarfi();
```

```
    iHarfi();
```

```
    XHarfi();
```

```
    CHarfi();
```

```
    VHarfi();
```

```

    BHarfi();
    NHarfi();
    MHarfi();
    ÖHarfi();
    ÇHarfi();
    Bosluk();
}

```

```
private void Form8_Load(object sender, EventArgs e)
```

```

{
    timer1.Interval = 3000;
    timer1.Start();
}

```

```
private void timer1_Tick(object sender, EventArgs e)
```

```

{
    bool bag = Connect("localhost", 6555);
    if (bag == true)
    {
        ListenerLoop();
    }
}

```

```
string JsonString = string.Empty;
```

```
ConvertJsonStringToDataTable jDt = new ConvertJsonStringToDataTable();
```

```
List<BtnKoordinat> eklenecek = new List<BtnKoordinat>();
```

```
private List<HeatPointButtonFormYena> HeatPoints = new
List<HeatPointButtonFormYena>();
```

```
public double konumAX, konumAY, konumAAX, konumAAY;
```

```
private async void ListenerLoop()
```

```

{

    StreamReader reader = new StreamReader(socket.GetStream());
    List<string> _harfListesi = new List<string>();
    isRunning = true;
}

```

```

double iX;
double iY;
double iXX;
double iYY;
while (isRunning)
{
    string response = string.Empty;

    try
    {
        response = await reader.ReadLineAsync();
        if (response != null)
        {
            textBox2.Text = response + Environment.NewLine;

            File file = new File();
            byte[] i = new byte[0];
            StringBuilder sbLog = new StringBuilder();
            sbLog.AppendLine(response);

            file.Write("C:\\Loglama\\", "Log6.txt", sbLog.ToString());

            foreach (DataRow dr in
jDt.JsonStringToDataTable(response.ToString()).Rows)
            {
                if (dr["category"].ToString() != "heartbeat")
                {
                    if(Convert.ToDouble(dr["lefteye"].ToString().Replace("avg:x:",
"".Replace(".", ""))< Convert.ToDouble(dr["righteye"].ToString().Replace("avg:x:",
"".Replace(".", ""))))
                {

```

```

        iX = Convert.ToDouble(dr["lefteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
        iXX = Convert.ToDouble(dr["righteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
    }
    else
    {
        iX = Convert.ToDouble(dr["righteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
        iXX = Convert.ToDouble(dr["lefteye"].ToString().Replace("avg:x:",
"").Replace(".", ","));
    }

    if(Convert.ToDouble(dr["lefteye y"].ToString().Replace(".", ","))<
Convert.ToDouble(dr["righteye y"].ToString().Replace(".", ",")))
    {
        iY = Convert.ToDouble(dr["lefteye y"].ToString().Replace(".", ","));
        iYY = Convert.ToDouble(dr["righteye y"].ToString().Replace(".", ","));
    }
    else
    {
        iY = Convert.ToDouble(dr["righteye y"].ToString().Replace(".", ","));
        iYY = Convert.ToDouble(dr["lefteye y"].ToString().Replace(".", ","));
    }
    for (int OO = 0; OO < eklenecek.Count; OO++)
    {
        if (((Convert.ToInt32(iX) >= eklenecek[OO].ButtonX) &&
(Convert.ToInt32(iXX) <= eklenecek[OO].ButtonXX)) && ((Convert.ToInt32(iY) >=
eklenecek[OO].ButtonY) && (Convert.ToInt32(iYY) <= eklenecek[OO].ButtonYY)))
        {
            _harfListesi.Add(eklenecek[OO].ButtonAdi);
            var gelen = _harfListesi.Last();
            if(gelen== eklenecek[OO].ButtonAdi)

```



```

        {
            if(HarfEkleme(eklenecek[OO].ButtonAdi,
DateTime.UtcNow.AddHours(+3).ToString("yyyy-MM-dd HH:mm:ss.fff")) >0)
                {
                    textBox1.Text += eklenecek[OO].ButtonAdi;
                    String filename = "C:\\Windows\\Media\\tada.wav";
                    System.Media.SoundPlayer sp = new
System.Media.SoundPlayer(filename);
                    sp.PlaySync();
                }
                // break;
            }
        }
        //break;
    }
    //break;
}
break;
}
}
catch (Exception ex)
{
    Console.Out.WriteLine("Error while reading response: " + ex.Message);
}
// DistinctValues(_harfListesi);
}
}

BtnKoordinat koordinat;

void AHarfi()
{
    koordinat = new BtnKoordinat();
}

```

```

        Point konumA = new Point();

        button13.Width = 100;

        button13.Height = 110;

        konumA.X = button13.Location.X ;

        konumA.Y = button13.Location.Y;

        konumAAX = konumA.X + button13.Width;

        konumAAY = konumA.Y + button13.Height;

        koordinat.ButtonAdi = button13.Text;

        koordinat.ButtonX =(int)konumA.X;

        koordinat.ButtonY = (int)konumA.Y;

        koordinat.ButtonXX = (int)konumAAX;

        koordinat.ButtonYY = (int)konumAAY;

        eklenecek.Add(koordinat);
    }

```

```

void SHarfi()

```

```

{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button14.Width = 100;

    button14.Height = 110;

    konumA.X = button14.Location.X;

    konumA.Y = button14.Location.Y;

    konumAAX = konumA.X + button14.Width;

    konumAAY = konumA.Y + button14.Height;

```

```

koordinat.ButtonAdi = button14.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void DHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button15.Width = 100;

    button15.Height = 110;

    konumA.X = button15.Location.X;

    konumA.Y = button15.Location.Y;

    konumAAX = konumA.X + button15.Width;

    konumAAY = konumA.Y + button15.Height;

    koordinat.ButtonAdi = button15.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);

}

```

```

void FHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button16.Width = 100;

    button16.Height = 110;

    konumA.X = button16.Location.X;

    konumA.Y = button16.Location.Y;

    konumAAX = konumA.X + button16.Width;

    konumAAY = konumA.Y + button16.Height;

    koordinat.ButtonAdi = button16.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);
}

```

```

void GHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button17.Width = 100;

    button17.Height = 110;

    konumA.X = button17.Location.X;

```

```

konumA.Y = button17.Location.Y;

konumAAX = konumA.X + button17.Width;

konumAAY = konumA.Y + button17.Height;

koordinat.ButtonAdi = button17.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);
}

void HHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button18.Width = 100;

    button18.Height = 110;

    konumA.X = button18.Location.X;

    konumA.Y = button18.Location.Y;

    konumAAX = konumA.X + button18.Width;

    konumAAY = konumA.Y + button18.Height;

    koordinat.ButtonAdi = button18.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

```

```

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);
}

void JHarfi()
{
koordinat = new BtnKoordinat();

Point konumA = new Point();

button19.Width = 100;

button19.Height = 110;

konumA.X = button19.Location.X;

konumA.Y = button19.Location.Y;

konumAAX = konumA.X + button19.Width;

konumAAY = konumA.Y + button19.Height;

koordinat.ButtonAdi = button19.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);
}

void KHarfi()
{

koordinat = new BtnKoordinat();

```

```

Point konumA = new Point();

button20.Width = 100;

button20.Height = 110;

konumA.X = button20.Location.X;

konumA.Y = button20.Location.Y;

konumAAX = konumA.X + button20.Width;

konumAAY = konumA.Y + button20.Height;

koordinat.ButtonAdi = button20.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);
}

void LHarfi()

{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button21.Width = 100;

    button21.Height = 110;

    konumA.X = button21.Location.X;

    konumA.Y = button21.Location.Y;

    konumAAX = konumA.X + button21.Width;

    konumAAY = konumA.Y + button21.Height;

```

```

koordinat.ButtonAdi = button21.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void ŞHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button22.Width = 100;

    button22.Height = 110;

    konumA.X = button22.Location.X;

    konumA.Y = button22.Location.Y;

    konumAAX = konumA.X + button22.Width;

    konumAAY = konumA.Y + button22.Height;

    koordinat.ButtonAdi = button22.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);

}

```



```
void iHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button23.Width = 100;

    button23.Height = 110;

    konumA.X = button23.Location.X;

    konumA.Y = button23.Location.Y;

    konumAAX = konumA.X + button23.Width;

    konumAAY = konumA.Y + button23.Height;

    koordinat.ButtonAdi = button23.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);
}
```

```
void XHarfi()
{
```

```
koordinat = new BtnKoordinat();

Point konumA = new Point();

button25.Width = 100;

button25.Height = 110;

konumA.X = button25.Location.X;

konumA.Y = button25.Location.Y;

konumAAX = konumA.X + button25.Width;

konumAAY = konumA.Y + button25.Height;

koordinat.ButtonAdi = button25.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}
```

```
void CHarfi()
```

```
{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button26.Width = 100;

    button26.Height = 110;

    konumA.X = button26.Location.X;

    konumA.Y = button26.Location.Y;

    konumAAX = konumA.X + button26.Width;
```

```

konumAAY = konumA.Y + button26.Height;

koordinat.ButtonAdi = button26.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);
}

void VHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button27.Width = 100;

    button27.Height = 110;

    konumA.X = button27.Location.X;

    konumA.Y = button27.Location.Y;

    konumAAX = konumA.X + button27.Width;

    konumAAY = konumA.Y + button27.Height;

    koordinat.ButtonAdi = button27.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);
}

```

```

}

void BHarfi()

{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button28.Width = 100;

    button28.Height = 110;

    konumA.X = button28.Location.X;

    konumA.Y = button28.Location.Y;

    konumAAX = konumA.X + button28.Width;

    konumAAY = konumA.Y + button28.Height;

    koordinat.ButtonAdi = button28.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);
}

void NHarfi()

{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button29.Width = 100;

    button29.Height = 110;

```

```

konumA.X = button29.Location.X;

konumA.Y = button29.Location.Y;

konumAAX = konumA.X + button29.Width;

konumAAY = konumA.Y + button29.Height;

koordinat.ButtonAdi = button29.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);
}

void MHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button30.Width = 100;

    button30.Height = 110;

    konumA.X = button30.Location.X;

    konumA.Y = button30.Location.Y;

    konumAAX = konumA.X + button30.Width;

    konumAAY = konumA.Y + button30.Height;

    koordinat.ButtonAdi = button30.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

```

```

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void ÖHarfi()

{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button31.Width = 100;

    button31.Height = 110;

    konumA.X = button31.Location.X;

    konumA.Y = button31.Location.Y;

    konumAAX = konumA.X + button31.Width;

    konumAAY = konumA.Y + button31.Height;

    koordinat.ButtonAdi = button31.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);

}

void ÇHarfi()

{

```

```
koordinat = new BtnKoordinat();

Point konumA = new Point();

button32.Width = 100;

button32.Height = 110;

konumA.X = button32.Location.X;

konumA.Y = button32.Location.Y;

konumAAX = konumA.X + button32.Width;

konumAAY = konumA.Y + button32.Height;

koordinat.ButtonAdi = button32.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}
```

```
void QHarfi()
```

```
{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button1.Width = 100;

    button1.Height = 110;

    konumA.X = button1.Location.X;

    konumA.Y = button1.Location.Y;

    konumAAX = konumA.X + button1.Width;
```

```
konumAAY = konumA.Y + button1.Height;
```

```
koordinat.ButtonAdi = button1.Text;
```

```
koordinat.ButtonX = (int)konumA.X;
```

```
koordinat.ButtonY = (int)konumA.Y;
```

```
koordinat.ButtonXX = (int)konumAAX;
```

```
koordinat.ButtonYY = (int)konumAAY;
```

```
eklenecek.Add(koordinat);
```

```
}
```

```
void ZHarfi()
```

```
{
```

```
koordinat = new BtnKoordinat();
```

```
Point konumA = new Point();
```

```
button24.Width = 100;
```

```
button24.Height = 110;
```

```
konumA.X = button24.Location.X;
```

```
konumA.Y = button24.Location.Y;
```

```
konumAAX = konumA.X + button24.Width;
```

```
konumAAY = konumA.Y + button24.Height;
```

```
koordinat.ButtonAdi = button24.Text;
```

```
koordinat.ButtonX = (int)konumA.X;
```

```
koordinat.ButtonY = (int)konumA.Y;
```

```
koordinat.ButtonXX = (int)konumAAX;
```



```

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void WHarfi()

{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button2.Width = 100;

    button2.Height = 110;

    konumA.X = button2.Location.X;

    konumA.Y = button2.Location.Y;

    konumAAX = konumA.X + button2.Width;

    konumAAY = konumA.Y + button2.Height;

    koordinat.ButtonAdi = button2.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);

}

void EHarfi()

{

```

```
koordinat = new BtnKoordinat();

Point konumA = new Point();

button3.Width = 100;

button3.Height = 110;

konumA.X = button3.Location.X;

konumA.Y = button3.Location.Y;

konumAAX = konumA.X + button3.Width;

konumAAY = konumA.Y + button3.Height;

koordinat.ButtonAdi = button3.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}
```

```
void RHarfi()
```

```
{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button4.Width = 100;

    button4.Height = 110;

    konumA.X = button4.Location.X;

    konumA.Y = button4.Location.Y;
```

```

konumAAX = konumA.X + button4.Width;

konumAAY = konumA.Y + button4.Height;

koordinat.ButtonAdi = button4.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void THarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button5.Width = 100;

    button5.Height = 110;

    konumA.X = button5.Location.X;

    konumA.Y = button5.Location.Y;

    konumAAX = konumA.X + button5.Width;

    konumAAY = konumA.Y + button5.Height;

    koordinat.ButtonAdi = button5.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

```

```

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void YHarfi()

{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button6.Width = 100;

    button6.Height = 110;

    konumA.X = button6.Location.X;

    konumA.Y = button6.Location.Y;

    konumAAX = konumA.X + button6.Width;

    konumAAY = konumA.Y + button6.Height;

    koordinat.ButtonAdi = button6.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);

}

void UHarfi()

{

```

```
koordinat = new BtnKoordinat();

Point konumA = new Point();

button7.Width = 100;

button7.Height = 110;

konumA.X = button7.Location.X;

konumA.Y = button7.Location.Y;

konumAAX = konumA.X + button7.Width;

konumAAY = konumA.Y + button7.Height;

koordinat.ButtonAdi = button7.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}
```

```
void IHarfi()
```

```
{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button8.Width = 100;

    button8.Height = 110;

    konumA.X = button8.Location.X;

    konumA.Y = button8.Location.Y;
```

```

konumAAX = konumA.X + button8.Width;

konumAAY = konumA.Y + button8.Height;

koordinat.ButtonAdi = button8.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void OHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button9.Width = 100;

    button9.Height = 110;

    konumA.X = button9.Location.X;

    konumA.Y = button9.Location.Y;

    konumAAX = konumA.X + button9.Width;

    konumAAY = konumA.Y + button9.Height;

    koordinat.ButtonAdi = button9.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

```

```

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

private void button1_Click(object sender, EventArgs e)
{
    Point o= FindLocation(button13);
    MessageBox.Show(o.X + " "+o.Y);
}

void PHarfi()
{
    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button10.Width = 100;

    button10.Height = 110;

    konumA.X = button10.Location.X;

    konumA.Y = button10.Location.Y;

    konumAAX = konumA.X + button10.Width;

    konumAAY = konumA.Y + button10.Height;

    koordinat.ButtonAdi = button10.Text;

    koordinat.ButtonX = (int)konumA.X;

```

```

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void ĞHarfi()

{

    koordinat = new BtnKoordinat();

    Point konumA = new Point();

    button11.Width = 100;

    button11.Height = 110;

    konumA.X = button11.Location.X;

    konumA.Y = button11.Location.Y;

    konumAAX = konumA.X + button11.Width;

    konumAAY = konumA.Y + button11.Height;

    koordinat.ButtonAdi = button11.Text;

    koordinat.ButtonX = (int)konumA.X;

    koordinat.ButtonY = (int)konumA.Y;

    koordinat.ButtonXX = (int)konumAAX;

    koordinat.ButtonYY = (int)konumAAY;

    eklenecek.Add(koordinat);

}

```



```

void ÜHarfi()
{
koordinat = new BtnKoordinat();

Point konumA = new Point();

button12.Width = 100;

button12.Height = 110;

konumA.X = button12.Location.X;

konumA.Y = button12.Location.Y;

konumAAX = konumA.X + button12.Width;

konumAAY = konumA.Y + button12.Height;

koordinat.ButtonAdi = button12.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

void Bosluk()
{

```

```

koordinat = new BtnKoordinat();

Point konumA = new Point();

button33.Width = 996;

button33.Height = 72;

konumA.X = button33.Location.X;

konumA.Y = button33.Location.Y;

konumAAX = konumA.X + button33.Width;

konumAAY = konumA.Y + button33.Height;

koordinat.ButtonAdi = button33.Text;

koordinat.ButtonX = (int)konumA.X;

koordinat.ButtonY = (int)konumA.Y;

koordinat.ButtonXX = (int)konumAAX;

koordinat.ButtonYY = (int)konumAAY;

eklenecek.Add(koordinat);

}

}

```

```

public struct HeatPointButtonFormYena

```

```

{
    public double X;
    public double Y;
    public double XX;
    public double YY;

    public HeatPointButtonFormYena(double iX, double iY, double iXX, double iYY)
    {
        X = iX;

```

```

    Y = iY;
    XX = iXX;
    YY = iYY;
}
}

public int HarfEklemeK(string mesaj,string tarih)
{
    SqlConnection baglan = new SqlConnection("Data Source=DESKTOP-
FG7D15E\\SQLEXPRESS;Initial Catalog=OkulTez;Integrated Security=True");
    SqlCommand _cmd = new SqlCommand("EklemeK", baglan);
    _cmd.CommandType = CommandType.StoredProcedure;
    _cmd.Parameters.Add(new SqlParameter("@harf", mesaj));
    _cmd.Parameters.Add(new SqlParameter("@zaman", tarih ));

    try
    {
        baglan.Open();
        int donen= _cmd.ExecuteNonQuery();
        baglan.Close();
        baglan.Dispose();
        return donen;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
        return 0;
    }
}
}

```