

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİYOLOJİK VERİ BİLİMİ

KURU GÖZ SENDROMUNA PROGRAMLAMA TEMELLİ
TANISAL SEÇENEK: GERÇEK ZAMANLI OTOMATİK GÖZ
KIRPMA SAYACI

YÜKSEK LİSANS TEZİ

ALİ RIZA CENK ÇELEBİ

HAZİRAN 2022

T.C.
İSTANBUL MEDENİYET ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİYOLOJİK VERİ BİLİMİ

KURU GÖZ SENDROMUNA PROGRAMLAMA TEMELLİ
TANISAL SEÇENEK: GERÇEK ZAMANLI OTOMATİK GÖZ
KIRPMA SAYACI

YÜKSEK LİSANS TEZİ

ALİ RIZA CENK ÇELEBİ

DANIŞMAN

PROF. DR. HANDAN ANKARALI

EŞ-DANIŞMAN

PROF. DR. HASAN GÜÇLÜ

HAZİRAN 2022

BİLDİRİM

Hazırladığım tezin tamamen kendi çalışmam olduğunu, akademik ve etik kuralları gözeterek çalıştığımı ve her alıntıya kaynak gösterdiğimi taahhüt ederim

Dr. Ali Rıza Cenk Çelebi

Danışmanlığını yaptığım işbu tezin tamamen öğrencinin çalışması olduğunu, akademik ve etik kuralları gözeterek çalıştığımı taahhüt ederim.

Prof. Dr. Handan Ankaralı

ONAY

Ali Rıza Cenk ÇELEBİ tarafından hazırlanan ‘Kuru Göz Sendromuna Programlama Temelli Tanısal Seçenek: Gerçek Zamanlı Otomatik Göz Kırpma Sayacı’ başlıklı bu yüksek lisans tezi, Biyolojik Veri Bilimi Bilim Dalında hazırlanmış ve jürimiz tarafından kabul edilmiştir.

JÜRİ ÜYELERİ

İMZA

Tez Danışmanı:

Prof. Dr. Handan ANKARALI

Kurumu: İstanbul Medeniyet Üniversitesi

Tez Eş-Danışmanı:

Prof. Dr. Hasan GÜÇLÜ

Kurumu: İstanbul Medeniyet Üniversitesi

Üyeler:

Prof. Dr. Özay Öz

Kurumu: İstanbul Yeni Yüzyıl Üniversitesi

Dr. Öğr. Ü. Erkan KARABEKMEZ

Kurumu: İstanbul Medeniyet Üniversitesi

Dr. Öğr. Ü. Özge PASİN

Kurumu: Bezmialem Vakıf Üniversitesi

Tez Savunma Tarihi: 24.06.2022

TEŞEKKÜR

Tez çalışmasını yakından takip ederek akademik ve deneyimlerini benimle paylaşan, tezin her aşamasında bilimsel katkı, ilgi ve desteğini hiçbir zaman esirgemeyen, sabırla yönlendiren, kendisinden çok şey öğrendiğim değerli danışman hocam, Sayın Prof. Dr. Handan ANKARALI'ya çok teşekkür ederim.

Fedakârlıkları ve bana olan güveni ile tez çalışmam boyunca her zaman yanımda olan aileme, arkadaşlarıma ve özellikle hayat eşim Özlem ÖNERCİ ÇELEBİ ile biricik kızım Elif ÇELEBİ'ye tüm kalbimle teşekkür ederim.

İÇİNDEKİLER

BİLDİRİM	i
ONAY	ii
TEŞEKKÜR	iii
KISALTMALAR	vi
TABLO LİSTESİ	vii
ŞEKİL LİSTESİ	viii
ÖZET	ix
ABSTRACT	x
BÖLÜM I (GİRİŞ)	1
BÖLÜM II (LİTERATÜR ÖZETİ)	3
2. LİTERATÜR ÖZETİ	4
2.1 KURU GÖZ SENDROMU TANIMI	4
2.2 KURU GÖZ SENDROMU TANISINI KOYMADA KULLANILAN YÖNTEMLER	4
2.3 GÖZ KIRPMA SAYISININ KURU GÖZ SENDROMUNDA ÖNEMİ	6
BÖLÜM III (MATERYAL VE METOT)	8
3. MATERYAL VE METOT	9
3.1 GÖRÜNTÜ İŞLEME	9
3.2 PYTHON PROGRAMLAMA DİLİ	11
3.3 GÖRÜNTÜ İŞLEMEDE KULLANILAN PYTHON KÜTÜPHANELERİ	13
3.4 OPENCV	15
3.5 GÖZ KIRPMA SAYISININ OPENCV KÜTÜPHANESİ KULLANILARAK ÖLÇÜLMESİ	16
BÖLÜM IV (VERİ)	19
4.1 HASTA SEÇİMİ	19
4.2 İSTATİSTİKSEL ANALİZLER	20

BÖLÜM V(BULGULAR).....	21
5.1 TANIMLAYICI İSTATİSTİKLER ve GRUPLAR ARASINDAKİ FARKLAR.....	21
5.2 KURU GÖZ GRUBU VE SAĞLIKLI GRUPLARDA FARKLI TASKLARDAKİ GÖZ KIRPMA SAYILARININ KARŞILAŞTIRILMASI...24	
5.3 KONVASİYONEL TESTLER İLE GÖZ KIRPMA SAYISI DEĞERLERİNİN KORELASYONU.....	25
5.4 KONVASİYONEL TESTLER İLE GÖZ KIRPMA SAYISININ ÖNGÖRÜLEBİLİRLİĞİ- REGRESYON EŞİTLİKLERİ-.....	27
BÖLÜM VI (TARTIŞMA).....	28
6. TARTIŞMA.....	28
KAYNAKÇA.....	33
EK 1. GÖZ KIRPMA SAYISINI ÖLÇEN PYTHON KODU.....	39

KISALTMALAR

KGS: Kuru Göz Sendromu

VDTS: Video Görüntüleme Terminalleri

CVS: Bilgisayar Görme Sendromu

TFOS: Göz Yaşı Film Tabakası Oküler Cemiyeti

TBUT: Gözyaşı Kırılma Zamanı

OSDI: Oküler Yüzey Hastalık İndeksi

DEWS: Kuru Göz Çalıştay

OpenCV: Açık Kaynak Bilgisayarla Görü

ICC: Sınıf içi Korelasyon Katsayısı

TABLO LİSTESİ

Tablo 1. Kuru Göz Sendromu ve Normal Gruplardaki Dağılımın Tespiti için Gerçekleştirilen Normalite Testi Sonuçları	21
Tablo 2. Kuru Göz Sendromu ve Normal Bireylerdeki Tanımlayıcı İstatistik Sonuçları.....	23
Tablo 3. Kuru Göz Sendromu ve Normal Bireylerdeki Farklı Tasklardaki Göz Kırpma Sayısının Karşılaştırılması.....	24
Tablo 4. Schirmer testi için Sınıf içi Korelasyon Katsayısı (ICC)	25
Tablo 5. TBUT testi için Sınıf içi Korelasyon Katsayısı (ICC).....	25
Tablo 6. Python Kodu ile Ölçülen Göz Kırpma Sayısı için Sınıf içi Korelasyon Katsayısı (ICC)	25
Tablo 7. Python Kodu ile Ölçülen Göz Kırpma Sayılarının Konvansiyonel Testler ile Ölçülen Değerler ile Korelasyonun Değerlendirilmesi	26
Tablo 8. Konvansiyonel Testler ile Göz Kırpma Sayacı Test Sonuçlarının Öngörülebilirliğini İlişkilendiren Regresyon Eşitlikleri	27

ŞEKİL LİSTESİ

Şekil 1. OSDI Anketi	5
Şekil 2. Gerçek Zamanlı Video Görüntüsü Alınmasını Sağlayan Python Kodları.....	16
Şekil 3. Gerçek Zamanlı Videodan Göz Kırpma Sayısını Ölçümünü Sağlayan Python Kodları.....	18
Şekil 4. Gerçek Zamanlı Videodan Gözler Açıkken Ortaya Çıkan Göz ve Yüz Konturlarını Tanımlayan Python Kodları	18
Şekil 5. Gerçek Zamanlı Videoda Gözler Kapalıyken Ortaya Çıkan Yüz Konturları ve Çıktıda Göz Kırpma Sayısını Gösteren Python Kodları	19

ÖZET

Kuru Göz Sendromu (KGS), gözyaşı film tabakasının homeostaz kaybı ile karakterize, oküler semptomların eşlik ettiği oküler yüzeyin çok faktörlü bir hastalığıdır. Kuru göz sendromunda risk faktörleri olarak; yaş, çeşitli hormonal hastalıklar, sistemik kullanılan ilaçlar (akne tedavisinde kullanılan isotretionin gibi), sigara kullanımı, geçirilmiş lazer refraktif cerrahi, kontakt lens kullanımı, düşük çevresel nem ve özellikle son dönemlerde sıklıkla kullandığımız bilgisayar, TV gibi çeşitli video görüntüleme terminallerinin (VDTS) uzamış kullanımı sayılmaktadır. Hatta yeterli gözyaşı üretimi olsa bile uzamış VDTS kullanımıyla ilişkili olarak göz kırpmasının azalması neticesinde gözyaşı film tabakası yetersiz dağılmakta ve neticesinde kuru göz sendromu gelişebilmektedir. Bu tezin amacı, Phyton programlama dilinde oluşturulan bir yazılım ile kuru göz sendromlu hastaların göz kırpma sayısını hesaplamak ve sonuçları normal sağlıklı bireylerle karşılaştırmaktır. Sonuçlara göre kuru göz sendromunda göz kırpma sayısının arttığı bulunmuştur. Bunun dışında hem kuru göz sendromlu hastalarda hem de sağlıklı kişilerde basılı materyal okunurken hesaplanan göz kırpma sayısının aynı materyalin dijital ekranda okunurken hesaplanan göz kırpma sayısından daha yüksek olduğu da bu tezde gösterilmiştir. Yine bu çalışma ile objektif olarak göz kırpma miktarını ölçen bu kod vasıtasıyla elde edilen göz kırpma sayısı değerlerinin kuru göz tanısında kullanılan diğer konvansiyonel tanı metodlarındaki sayısal değerler ile de yüksek düzeyde korelasyon gösterdiği saptanmıştır.

Anahtar Kelimeler: Kuru Göz Sendromu, Konvansiyonel Tanı Tetkikleri, Göz Kırpma Sayısı, Dijital Ekran Kullanımı, Python Programlama Dili, Görüntü İşleme, OpenCV Kütüphanesi

ABSTRACT

Dry Eye Syndrome is a multifactorial disease of the ocular surface accompanied by ocular symptoms and is characterized by the loss of homeostasis of the tear film layer. The risk factors for dry eye syndrome include age, various hormonal disorders, systemic medications (like isotretinoin for acne), smoking, history of refractive laser surgery, use of contact lenses, low humidity and extended use of computer, television and various video display terminal systems that has become more frequent recently. Decrease of blinking due to extended use of these devices may result in an inadequate distribution of the tear film layer and subsequently give rise to dry eye syndrome, even though there is enough production of tear. The aim of this thesis is to calculate the number of blinks of patients with dry eye syndrome with codes created in Python programming language and to compare the results with normal healthy individuals. According to the results, it was found that the number of blinks increased in dry eye syndrome. Other than that, this thesis also showed that the number of blinks calculated while reading form of printed material was higher than the one that was calculated while reading the same form on the digital screen, both in patients with dry eye syndrome and in healthy subjects. This thesis also showed that the number of blinks measured objectively using this code showed strong correlation with the conventional diagnostic methods used for diagnosis of dry eye syndrome.

Keywords: Dry Eye Syndrome, Conventional Diagnostic Methods, Eye Blink Count, Digital Screen Use, Python Programming Language, Image Processing, OpenCV Library

BÖLÜM I (GİRİŞ)

Kuru Göz Sendromu (KGS), gözyaşı film tabakasının homeostaz kaybı ile karakterize, oküler semptomların eşlik ettiği, gözyaşı film tabakasının dayanıksızlığı, hiperozmolaritesi, oküler yüzey enflamasyonu, hasarı ve nörosensöriyel anormalliklerin etyolojik olarak rol oynadığı, oküler yüzeyin çok faktörlü bir hastalığıdır. Kuru göz sendromunda risk faktörleri olarak; yaş, çeşitli hormonal hastalıklar, sistemik kullanılan ilaçlar (akne tedavisinde kullanılan isotretionin gibi), sigara kullanımı, geçirilmiş lazer refraktif cerrahi, kontakt lens kullanımı, düşük çevresel nem ve özellikle son dönemlerde sıklıkla kullandığımız bilgisayar, TV, telefon gibi çeşitli video görüntüleme terminallerinin (VDTS) uzamış kullanımını sayabiliriz. Yeterli gözyaşı üretimi olsa bile VDTS kullanımıyla ilişkili olarak göz kırpmasının azalması neticesinde gözyaşı film tabakası yetersiz dağılmakta ve neticesinde kuru göz sendromu gelişmektedir (Jaiswal ve diğ.,2019). Bu tablonun neticesi olarak refleks olarak kuru göz semptomlarının artışı ile yüksek göz kırpma sayısı gözlenmektedir. Ancak görsel aktiviteler ne yazık ki göz kırpma miktarını azaltmakta ve kuru göz semptomlarını kötüleştirmektedir (Cardona ve diğ., 2011). Günün sonunda VDTS kullanıcılarında göz kırpma sayısı belirgin azalmaktadır. Bu tablo ile de ilişkili olarak son dönemlerde bilgisayar görme sendromu olarak da dilimize çevrilen Computer Vision Syndrome (CVS), Amerikan Optometrik Derneği tarafından bilgisayar kullanımı ile ilişkili göz ve görme problemlerinin kombinasyonu olarak tanımlanmaktadır (Rosenfeld, 2011). Bu belirtiler, bireyin bilgisayar görevini rahatça yerine getirmek için yetersiz görsel yeteneklere sahip olmasından kaynaklanır. 2003 yılında işlerin %55'inin bilgisayar kullanımı içerdiği tahmin edilmiştir (Rosenfeld, 2011). Günümüzde bu sayı artmıştır. Gerçekten de yakın tarihli bir rapor, yetişkinlerin günde ortalama 8,5 saat elektronik ekran görüntüleyebileceğini ileri sürmüştür (Rosenfeld, 2011). İşte bu nedenledir ki Amerikan Optometri derneği 20 dakikada bir bireylerin gözünü VDTS ekranlarından ayırmalarını önermektedir. Bu bilgiler ışığında göz kırpmasının neden önemli bir değişken olduğunun sorgulanması gündeme gelmiştir. Göz kırpmak oküler yüzey parametrelerini etkilemektedir (Wang

ve diğ., 2018). Rutin olarak normal bir kuru göz sendromunda yapılan testlerin değerleri göz kırpma sayısı ile düzelebilmekte, dolayısı ile göz kırpma sayısının doğru bir şekilde belirlenmesi ve hatta uyarılmasının sağlanması kuru göz sendromunda tedavi edici bir role dahi sahip olabilmektedir.

Rutin oftalmoloji pratiğinde kuru göz sendromu şüphesiyle kliniğe başvuran hastalarımıza; biyomikroskopide erken gözyaşı kırılma zamanı ((TBUT) <10sn), Oküler Yüzey Hastalık İndeksi (OSDI) olarak da bilinen hasta anketleri (>12 kuru göz olarak kabul edilmekte), Schirmer testi (<10 mm/5dk değeri kuru göz ile ilişkilidir) gibi testler uygulanmaktadır (Craig ve diğ.,2017).

Kuru göz sendromu semptomatik bir tanıdır (%40 asemptomatik) ve yukarıda bahsi geçen testlerden elde edilen veriler ile de kesin tanı konulur. Ancak görüldüğü üzere bu testler arasında göz kırpma sayısı yoktur. Bir önceki bölümde de önemi belirtilen göz kırpma sayısının özellikle son on yılda artan VDTS kullanımı ile ölçülmesinin gerekliliği aşikardır. Bu kapsamda tamamen ücretsiz ve açık kaynak kodlu öğrenilmesi kolay zengin bir modül (kütüphane) kaynağına sahip python diliyle yazılan yazılım ile göz kırpma sayısı bu tezde ölçülmüştür. Bu yazılım dili altında da kolayca çalışan görüntü işleme kütüphanelerinden birisi olan OpenCV kütüphanesi ile gerçek zamanlı videodan göz kırpma sayısı yazdığımız kod ile kolaylıkla gerçekleştirilebilmektedir. Bu tez çalışmasında OpenCV kütüphanesinin kullanımı ile oluşturulan kod; normal ve kuru göz semptomları olan hastalarda çeşitli aktivitelerde (task 1= 5 dakika boyunca bilgisayar ekranından bir A4 sayfa metin okuma, task 2= 5 dakika boyunca bir A4 kâğıda basılı metin okuma) farklı dikkat seviyelerindeki göz kırpma miktarları ölçülmüştür. Yine bu tez kapsamında elde edilen göz kırpma sayısı değerleri ile kuru göz sendromu tanısında kullandığımız Schirmer testi, TBUT testi ile OSDI skorlarının korelasyonuna da bakılmıştır. Bu tez içeriğinde Acıbadem Atakent Eğitim Araştırma Hastanesi Göz Kliniğine göz kuruluğu şikayetleri ile gelen ve konvansiyonel tanı tetkikleri olan TBUT <10sn, OSDI>12 ve Schirmer testi <10 mm / 5dk olan bireylere kuru göz sendromu tanısı konulmuş ve bu grup kuru göz grubu olarak alınmış, bu bulgulardan herhangi bir tanesine sahip olmayan göz kliniğine rutin gözlük muayenesine gelen sistemik hastalığı olmayan ve herhangi bir göz damlası kullanmayan yaş ile cinsiyet olarak kuru göz grubuna benzer özellikler gösteren bireyler de sağlıklı kontrol grubu olarak çalışmaya dahil edilmişlerdir.

Bu tezin ana katkıları şu şekilde özetlenebilir:

Kuru göz sendromu tanısı konvansiyonel tetkikler ile konan hastaların göz kırpması sayısı Python programlama kullanılarak oluşturulmuş kodlar vasıtasıyla sayılmış ve elde edilen bu sonuçlar sağlıklı bireylerdeki değerler ile karşılaştırılmıştır. Ayrıca objektif olarak göz kırpması miktarını ölçen bu kod vasıtasıyla elde edilen değerlerin kuru göz tanısında kullanılan diğer konvansiyonel tanı metodlarındaki sayısal değerler ile korelasyonuna da bakılmıştır. Bu tezin ikinci bölümünde literatür taraması kapsamında kuru göz sendromu tanımı ile kuru göz tanısı koymada kullanılan çeşitli yöntemler ile göz kırpması sayısının kuru göz tanısındaki önemi tartışılmıştır. Bölüm 3'te göz kırpmasının ölçüleceği gerçek zamanlı görüntülerin işlenmesi üzerine görüntü işleme başlığı altında genel bilgiler verilecek, bu kapsamda teze konu olan OpenCV kütüphanesi ile python programlama dili hakkında bilgiler sunulmuştur. Ayrıca görüntü işleme sırasında kullanılan yapay zekâ algoritmaları hakkında da bilgiler paylaşılmıştır. Bölüm 4'te çalışma kapsamında kullanılan veri kümelerinden ve tezde değerlendirme amaçlı kullanılan istatistiksel analizler sunulmaktadır. Bölüm 5'te kullanılan yöntemler ile yapılan deneylerden elde edilen sonuçlar yer almaktadır. Bölüm 6'da yapılan tez çalışmasının sonuç değerlendirmesi sunulmuştur.

BÖLÜM II (LİTERATÜR ÖZETİ)

Kuru Göz Sendromu tanımı, tanıda kullanılan yöntemler ile göz kırpması sayısının kuru göz sendromundaki önemi ile ilgili kavramsal bilgilere bu bölümde yer verilmektedir.

2. LİTERATÜR ÖZETİ

2.1 KURU GÖZ SENDROMU TANIMI

Son otuz yılda, dünya çapında kuru göz sendromu (KGS) farkındalığının önemli ölçüde arttığı görülmüştür. Küresel olarak, KGS prevalansı, KGS'nun tanımına, nüfusun yaş dağılımına ve metodolojiye bağlı olarak geniş bir aralıkta değişmektedir. Kuru göz hastalığı prevalansı toplum temelli epidemiyolojik çalışmalarda %5,5 ile %50,1 arasında ve bazı hastane temelli çalışmalarda %7,99 ile %29,9 arasında değişmektedir (Yılmaz ve diğ., 2015). Etkilenen bireyler için klinik bakımı iyileştirmeye yönelik devam eden girişimlerde, birçok kuruluşun karşılıklı çabalarıyla, bu hastalığın temeli ve etkisi hakkında çok şey öğrenilmiştir. Göz Yaşı Film Tabakası Oküler Cemiyeti (TFOS) Tanımlama ve Sınıflandırma Alt Komitesi tarafından tanım gözden geçirilmiş, kanıta dayalı bir kuru göz tanımına ve KGS'nun kolektif mevcut anlayışıyla tutarlı bir sınıflandırma şemasına ulaşılmıştır. Kuru göz için önceki tanımlar ve sınıflandırma şemaları başlangıç noktasını oluşturmuş ve mevcut literatürden elde edilen en son bilgiler bağlamında ve yanıtları ile görüşleri dikkate alan fikir birliğine dayalı bir şekilde algılanan eksiklikleri gidermek için revizyonlar yapılmıştır. Kuru göz için küresel bir fikir birliğinin değerini kabul eden Tanımlama ve Sınıflandırma alt komitesi, daha geniş TFOS Kuru Göz Çalıştay (DEWS) II üye 23 ülkenin katkılarıyla evrensel olarak anlaşılır ve çevrilebilir bir tanım oluşturmaya çalışmıştır. Bu çalıştay sonucunda güncel olarak kuru göz sendromu; gözyaşı filmi homeostazının kaybı ile karakterize ve oküler semptomların eşlik ettiği, gözyaşı filmi instabilitesi ve hiperosmolaritesi, oküler yüzey iltihabı ve hasarı ve nörosensoriyel anormalliklerin etiyolojik rol oynadığı oküler yüzeyin çok faktörlü bir hastalıktır (Craig ve diğ.,2017).

2.2 KURU GÖZ SENDROMU TANISINI KOYMADA KULLANILAN YÖNTEMLER

Bu bölüm, özellikle önceki TFOS DEWS raporundan bu yana, Kuru göz sendromu (KGS) tanı ölçütlerini gözden geçirmektedir. KGS'nun önceki TFOS DEWS tanımında (DEWS grubu, 2007) olduğu gibi, KGS için mevcut TFOS DEWS II tanımı, oküler yüzey semptomlarının ve KGS'nun diğer belirtilerinin varlığını tanımlar (Craig ve diğ., 2017). KGS semptomları ve belirtileri arasındaki ilişki doğrusal olmamasına ve bireylere göre değişmesine rağmen (Begley ve diğ., 2003), oküler yüzey

semptomlarını doğru bir şekilde ölçme yeteneği, ek KGS değerlendirmesi için tıbbi gerekliliğin belirlenmesine yardımcı olabilecek önemli bir tarama aracıdır. Ayrıca bu tanısal ölçütler hastalık durumunun ilerlemesini ve tedavilere yanıtı izlemek için de kritik öneme sahiptir. Bu bağlamda, semptom ölçümleri KGS'nun klinik belirtilerine çok benzer. Bu nedenle, hasta etkileşiminin başlangıcında doğrulanmış bir semptom anketinin uygulanması tavsiye edilir. Buradan yola çıkılarak da Oküler Yüzey Hastalık Endeksi (OSDI), KGS klinik deneyleri için en yaygın kullanılan ankettir. OSDI kuru göz sendromu ile ilişkili semptomların sıklığını, çevresel tetikleyicileri ve görme ile ilgili yaşam kalitesini ölçer. OSDI anketi, görme bozukluğu (bulanık görme veya zayıf görme) veya görsel işlev (okuma, gece araba kullanma, bilgisayarda çalışma veya TV izleme sorunları) ile ilgili 6 soru içerir (Şekil 1).

Ek: OSDI anketi					
OSDI skoru					
Geçen hafta boyunca aşağıdakilerden herhangi birini yaşadınız mı?					
Her zaman	Çoğunlukla	Zamanın yarısında	Bazen	Hiçbir Zaman	
1- Gözlerde ışığa hassasiyet					
2- Gözlerde batma hissi					
3- Gözlerde ağrı ya da kızama					
4- Bulanık görme					
5- Az görme	(4)	(3)	(2)	(1)	(0)
					1-5 nolu sorulara verilen cevapların toplamı: (A)
Geçen hafta boyunca götüneünüzdeki problemler aşağıdaki aktivitelerinizi engelledi mi?					
Her zaman	Çoğunlukla	Zamanın yarısında	Bazen	Hiçbir Zaman	
6- Okuma					N/A
7- Gece araba kullanma					N/A
8- Bilgisayarda çalışma					N/A
9- Televizyon izleme					N/A
(4)	(3)	(2)	(1)	(0)	
N/A: Herhangi bir gözlem olmadığında işaretlenmelidir.					
6-9 nolu sorulara verilen cevapların toplamı: (B)					
Geçen hafta boyunca aşağıdaki durumlarla götüneünüzde rahatsızlık hissettiniz mi?					
Her zaman	Çoğunlukla	Zamanın yarısında	Bazen	Hiçbir Zaman	
10- Rüzgârda					N/A
11- Düşük nemli (çok kuru) yerlerde					N/A
12- Klimalı yerlerde					N/A
(4)	(3)	(2)	(1)	(0)	
N/A: Herhangi bir gözlem olmadığında işaretlenmelidir.					
10-12 nolu sorulara verilen cevapların toplamı: (C)					
(N/A olarak cevaplanan sorular eklenmeyecek)					
A+B+C = D					
Cevaplanan soru sayısı = E					
OSDI = (D X 25) / E					

Şekil 1. OSDI Anketi

Bir çalışma, 87 hastadan oluşan KGS grubunun, KGS olmayan 71 hastadan oluşan bir gruba karşılaştırıldığında, görme ile ilgili işlev için OSDI bileşik ve alt ölçek puanlarının daha kötü olduğunu göstermiştir (Li ve diğ., 2012). Görme bozukluğu şu anda oküler semptomoloji anketleri aracılığıyla subjektif olarak değerlendirilmektedir. Görme bozukluğunun iyi bilinen objektif klinik ölçümleri yaygın olarak kullanılabilmektedir.

kadar, KGS tanısı için TFOS DEWS II tarafından önerilebilecek özel bir ek görme testi yoktur. Ancak göz yaşı film stabilitesinin bozulması TFOS DEWS II tarafından dikkate alınan bir tanısal araçtır. Bozulmuş gözyaşı filmi stabilitesi, gözyaşı filminin anormalliğini teşhis etmek için temel tanı kriterlerinden biri olmuştur ve bu kapsamda gözyaşı filmi stabilitesini değerlendirmenin birçok yolu tarif edilmiştir (Sweeney ve diğ., 2013). Klinik uygulamada, gözyaşı filmi stabilitesinin değerlendirilmesinde en sık kullanılan test; gözyaşı filmi kırılma zamanının (TBUT) ölçümüdür; bu, tam bir göz kırpma ile gözyaşı filmindeki ilk kırılmanın ortaya çıkması arasında geçen zaman aralığıdır (Norn, 1969). Gözyaşı filminin görünürlüğünü arttırmak için sodyum floresein damlatılabilir. Gözyaşı filmi kırılma zamanının KGS teşhisi için referans değeri, 10 s'den daha kısa olacak şekilde ifade edilmektedir (Lemp ve Hamill, 1973). KGS tanımı içinde doğrudan belirtilmemesine rağmen, gözyaşı filmi hacmi oküler yüzey sağlığı için önemlidir ve homeostaz kaybı aynı zamanda KGS hastalarında buharlaşmadan bağımsız olarak önemli bir patojenik mekanizma ve tanısal bir işaret olabilir. Bu kapsamda klinik pratikte sıklıkla uygulanan anestezişiz Schirmer testi, şiddetli aköz eksikliği ile beraber giden KGS tanısını doğrulamak için önerilen bir tanı testi olmaya devam etmektedir (Vitali ve diğ., 2003). Schirmer testi, Schirmer kâğıt şeridinin (5 -35 mm) çentikten katlanması ve katlanmış ucun alt kapak kenarının geçici üçte birine kancalanmasıyla gerçekleştirilir. Skor, 5 dakikalık bir sürenin ardından çentikten ölçülen ıslanma uzunluğudur. Anestezişiz Schirmer testi, uyarılmış refleks gözyaşı akışının bir tahminini sağlayan iyi standardize edilmiş bir testtir. 5 mm/5 dak ile 10 mm/5 dak (de Monchy ve diğ., 2011) arasında birkaç tanısal eşik değer bu skor için KGS tanısının konulması adına önerilmiştir.

2.3 GÖZ KIRPMA SAYISININ KURU GÖZ SENDROMUNDA ÖNEMİ

Gözyaşı filminin birincil işlevi, göz kırpma aralıkları sırasında konfor ve kaliteli görüşü korumaktır. Gözyaşı disfonksiyonu, gözyaşı dengesizliği ve tahriş semptomları ile bulanık ve dalgalı görme ile sonuçlanır. Okuma veya video görüntüleme terminallerinin (VDTS) kullanımı sırasında göz kırpma hızının azaldığı bildirilmiş olmasına rağmen, (Tsubota ve Nakamori., 1993.) bazı gözyaşı disfonksiyonu hastaları bu aktiviteler sırasında daha fazla göz kırpma rapor etmektedir. Her ne kadar önceki çalışmalar gözyaşı işlev bozukluğu olan gözlerde göz kırpma hızının arttığını (Kato ve diğ., 2019) bulsa da bu çalışmaların çoğu normal göz kırpma hızlarını herhangi bir

nedenle gözyaşı filmi işlev bozukluğu/kuru göz ile karşılaştırmıştır. Kuru göz sendromunda gözlenen gözyaşı disfonksiyonunda göz kırpmasının önemi özellikle son dekatta sıklıkla kullandığımız bilgisayar, TV gibi çeşitli video görüntüleme terminallerinin (VDTS) uzamış kullanımı ile gündeme gelmiş bir konudur.

Göz kırpma, optik performansın ve oküler yüzeyin sağlığının korunmasında hayati önem taşır. Göz kırpma eylemi göz yüzeyindeki kalıntıları temizler, mekanik koruma sağlar ve gözyaşı filmi yeniden oluşturur (Bron ve diğ., 2004). Ayrıca, göz kırpma, meibum dağılımı ve uygun bir gözyaşı filmi lipid tabakasının yeniden oluşturulması (Doane, 1981) için hayati öneme sahiptir.

Normal spontan göz kırpma hızının dakikada 10 ila 15 göz kırpma olduğu bildirilmektedir (Doughty, 2001). Göz kırpmalar arası aralık denekler arasında değişkendir, DED'de azalır ve suni gözyaşı damlatma ile artırılabilir (Pult ve diğ.,2013). Bununla birlikte, göz kırpma hızı, Parkinson hastalığı gibi sistemik hastalıklardan ve uzamış VDTS kullanımı (Tsubota ve Kanamori., 1993) gibi durumlardan da etkilenir.

Kortikal kontrole rağmen, oküler yüzey koşulları göz kırpma hızı ile ilişkilendirilmiştir. Göz kırpma hızı, gözyaşı filmi kalitesinden etkilenebilir (Yap M, 1991) çünkü kuru göz hastalarında göz kırpma hızının normal popülasyondan daha yüksek olduğu gösterilmiştir. Bu kapsamda Nakamori ve ark., normal bireylerde göz kırpma hızını 20,1/5 dk olarak buna karşın kuru göz hastalığı olan bireylerde ise 34,1/5 dk düzeyinde bir göz kırpma hızı kaydetmiştir (Nakamori ve ark.,1997). Bu değerler çalışmamızda elde ettiğimiz sonuçlara oldukça benzerlik göstermektedir. Bu sonuçlar oküler yüzey hasarının göz kırpma sayısının arttığını da destekler niteliktedir.

Bir başka bakış açısı ile göz kırpmasının yetersizliğinin kuru göz sendromuna giden süreçteki nedensellik rolü ise şu şekilde rasyonelize edilmektedir. Yeterli gözyaşı üretiminin olması durumunda bile VDTS kullanımıyla ilişkili olarak göz kırpmasının azalması neticesinde gözyaşı film tabakası yetersiz dağılmakta ve sonuçta kuru göz sendromu gelişmektedir (Jaiwal ve diğ., 2019). Buradan yola çıkarak da bir tepki olarak kuru göz ile ilgili şikayetleri artışı ile yüksek miktarda göz kırpma sayısı gözlenebilmektedir. Ancak vücumuzda gözlenen bu refleks cevaba karşın dikkat gerektiren görsel aktivitelerin artışı ne yazık ki göz kırpma miktarını azaltmaya devam

etmekte ve günün sonunda ciddi kuru göz semptomları ile sonuçlanabilmektedir. (Cardona ve diğ., 2011). Büyük ölçekli, nüfusa dayalı bir Japon çalışması, 102.582 orta yaş katılımcıda (40 ila 74 yaş arası) günlük saat VDTS kullanımı ile KGS teşhisi arasındaki ilişkiyi değerlendirmeye çalışmıştır. Bu çalışma sonucunda, düzenli VDTS kullanıcılarında özellikle günde 5 saati aşan kullanımlarda KGS teşhisi için bir risk faktörü olarak tanımlanmıştır (Hanyuda ve ark., 2020). Yine teknolojik olarak ileri bir düzeyde yaygın VDTS kullanımının gözlemlendiği Japonya'da günlük bilgisayar kullanan 3549 genç ve orta yaşlı ofis çalışanı (22 ila 60 yaş arası) üzerinde yapılan bir başka çalışmada da KGS gelişiminin genç bireylerde artan VDTS kullanımı ile ilişkili olduğu sonucuna varılmıştır (Uchino ve diğ., 2008). Bu çalışmalar birlikte, uzun süreli VDTS kullanımı KGS gelişimi için bir risk faktörü olarak tanımlamaktadır. Buradan yola çıkarak son dönemlerde bilgisayar görme sendromu olarak da dilimize çevrilen Computer Vision Syndrome (CVS), Amerikan Optometrik Derneği tarafından bilgisayar kullanımı ile ilişkili göz ve görme problemlerinin kombinasyonu olarak tanımlanmıştır (Rosenfeld, 2011). 2003 yılında günlük ofis işlerin %55'inin bilgisayar kullanımı içerdiği rapor edilmiştir. Günümüzde yetişkinlerin günde ortalama 8,5 saat elektronik ekran görüntüleyebileceğini ifade edilmiştir. Buradan yola çıkarak Amerikan Optometri derneği 20 dakikada bir bireylerin gözünü VDTS ekranlarından ayırmalarını önermektedir. Göz kırpması bu kadar önemli iken bunun tespit edilmesi de ayrıca özen gösterilmesi gereken bir durumdur. Göz kırpmalarının değerlendirilmesi için şu anda mevcut olan yöntemlerle ilgili olarak, elektrookülografi, elektromiyografi, kızılötesi yansıma ve manyetik arama bobinlerinin tümü rapor edilmiştir. Son yıllarda, hızlı bir kamera ile göz kırpmalarını noninvaziv ve kısa sürede değerlendirmek için yeni yöntemlerin geliştirilmesine devam edilmektedir (Kato ve diğ., 2019). Bu kapsamda programcılık alanı ile yeni düzeyde çalışmalar yapılmaya başlanmış ve göz kırpmasının programlama temelli objektif sayımı üzerine literatürde çok sınırlı yayın bulunmaktadır.

BÖLÜM III (MATERYAL VE METOT)

Bölüm 3'te göz kırpmasının ölçüldüğü gerçek zamanlı görüntülerin çalışıldığı görüntü işleme başlığı altında genel bilgiler verilecek bu kapsamda teze konu olan openCV kütüphanesi ve python programlama dili hakkında içerikler sunulacaktır. Ayrıca

görüntü işleme sırasında kullanılan yapay zekâ algoritmaları hakkında da bilgi verilmektedir. Tezin bu bölümü yukarıda bahsi geçen materyallerin kullanım metotlarını da içermektedir.

3. MATERYAL VE METOT

3.1 GÖRÜNTÜ İŞLEME

Görüntü işleme, gelişmiş bir görüntü elde etmek veya ondan bazı yararlı bilgiler çıkarmak için bir görüntü üzerinde bazı işlemleri gerçekleştirme yöntemidir. Girişin bir görüntü olduğu ve çıkışın görüntü veya o görüntüyle ilişkili özellikler olabileceği bir sinyal işleme türüdür. Günümüzde görüntü işleme hızla gelişen teknolojiler arasında yer almaktadır. Mühendislik ve bilgisayar bilimleri disiplinlerinde de temel araştırma alanını oluşturur (Petrou, Petrou, 2010).

Görüntü işleme temel olarak aşağıdaki üç adımı içerir: Görüntü elde etme araçları aracılığıyla görüntünün içe aktarılması; görüntüyü analiz etme ve manipüle etme; Görüntü analizine dayalı olarak sonucun değiştirilebileceği görüntü veya rapor çıktısı. Görüntü işleme, bir kişinin bir görüntünün kalitesini artırabileceği veya bir görüntüden uyarıcı bilgiler toplayabileceği ve sonraki şeyleri tahmin etmek için bir algoritmaya besleyebileceği bir yoldur (Chellappa, 1992).

Görüntü işleme, bir görüntüyü dijital forma dönüştürme ve ondan bazı yararlı bilgiler elde etmek için belirli işlemleri gerçekleştirme sürecidir. Görüntü işleme sistemi, önceden belirlenmiş belirli sinyal işleme yöntemlerini uygularken genellikle tüm görüntüleri 2B sinyaller olarak ele alır (Gimel' farb, Patrice, 2018).

Beş ana görüntü işleme türü vardır:

Görselleştirme - Resimde görünmeyen nesneleri bulma

Tanıma - Görüntüdeki nesneleri ayırt etme veya algılama

Keskinleştirme ve geri yükleme - Orijinal görüntüden gelişmiş bir görüntü oluşturma

Desen tanıma - Görüntüdeki nesnelerin etrafındaki çeşitli desenleri ölçme

Alma - Orijinal görüntüye benzeyen geniş bir dijital görüntü veritabanındaki görüntülere göz atın ve bu görüntüleri arama

Görüntü işlemenin basamakları ise şu şekilde özetlenebilir (Gonzales ve diğ., 2017):

Görüntü elde etme: Görüntü işlemedeki ilk adımdır. Bu adım, görüntü işlemede ön işleme olarak da bilinir. Görüntünün bir kaynaktan, genellikle donanım tabanlı bir kaynaktan alınmasını içerir.

Görüntü İyileştirme: Gizlenmiş bir görüntüdeki belirli ilgi çekici özellikleri ortaya çıkarma ve vurgulama işlemidir. Bu, parlaklığı, kontrastı vb. değiştirmeyi içerebilir.

Görüntü Restorasyonu: Bir görüntünün görünümünü iyileştirme işlemidir. Ancak, görüntü iyileştirmeden farklı olarak, görüntü restorasyonu belirli matematiksel veya olasılıksal modeller kullanılarak yapılır.

Renkli Görüntü İşleme: Dijital bir alanda bir dizi renk modelleme tekniğini içerir. Bu adım, dijital görüntülerin internet üzerinden önemli ölçüde kullanılması nedeniyle önem kazanmıştır.

Dalgacıklar ve Çoklu Çözünürlüklü İşleme: Dalgacıklar, görüntüleri çeşitli çözünürlük derecelerinde temsil etmek için kullanılır. Görüntüler, veri sıkıştırma ve piramidal gösterim için dalgacıklara veya daha küçük bölgelere bölünür.

Sıkıştırma: Bir görüntüyü kaydetmek için gereken depolamayı veya onu iletmek için gereken bant genişliğini azaltmak için kullanılan bir işlemidir. Bu, özellikle görüntü internette kullanım için olduğunda yapılır.

Morfolojik İşleme: Görüntüleri şekillerine göre dönüştürmek için bir dizi işleme işlemidir.

Segmentasyon: Görüntü işlemenin en zor adımlarından biridir. Bir görüntünün kendisini oluşturan parçalara veya nesnelere bölünmesini içerir.

Reprezentasyon ve Tanımlama: Segmentasyon işleminde bir görüntü bölgelere bölündükten sonra, her bölge daha sonraki bilgisayar işlemleri için uygun bir biçimde temsil edilir ve tanımlanır. Reprezentasyon, görüntünün özellikleri ve bölgesel özellikleriyle ilgilenir. Tanımlama, bir nesne sınıfını diğerinden ayırt etmeye yardımcı olan nicel bilgilerin çıkarılmasıyla ilgilenir.

Tanıma: Tanımlamaya dayalı olarak bir nesneye bir etiketin atanması sağlanır.

Bugün kullandığımız en yaygın görüntü işleme uygulamalarından biri yüz algılamadır. Makinenin, yüz şekli, gözler arasındaki mesafe gibi insan yüzlerinin belirli özellikleriyle eğitildiği derin öğrenme algoritmaları ile eğitildikten sonra makine insan yüzüne benzeyen bir görüntüdeki tüm yüz özelliklerini tespit etmeye başlayacaktır. Yüz algılama, bugünlerde çoğu sosyal medya uygulamasında bulunan güvenlik, biyometri ve hatta filtrelerde kullanılan önemli bir araçtır (O’Gorman ve diğ., 2008).

3.2 PYTHON PROGRAMLAMA DİLİ

Python, 1989 yılında Hollandalı programcı Guido van Rossum tarafından tasarlanan genel amaçlı bir programlama dilidir. Python, Python Yazılım Vakfı aracılığıyla koordine edilen geliştirme sayesinde ücretsiz ve açık kaynak kodlu olarak kullanılabilir (Internet 1). Python, son on yılda hızlı bir şekilde benimsenmiş ve şu anda en yaygın kullanılan programlama dillerinden biri olmuştur. Python, aşağıdakiler gibi hemen hemen tüm uygulama etki alanlarında kullanılan genel amaçlı nesne tabanlı bir programlama dildir; iletişim, web geliştirme ve grafik kullanıcı arayüzleri, oyun geliştirme, yapay zekâ ve veri bilimi vb. Python, yeni başlayanlar için uygundur ve en iyi bilgisayar bilimi programlarında programlamayı öğretmek için rutin olarak kullanılmaktadır. Python, özellikle bilim ve veri bilimi topluluklarında popülerdir. Finans ve bankacılık alanlarında Excel gibi tanınmış araçların yerini sürekli olarak alıyor. Python, dünya çapında en popüler programlama dillerinden biridir. Python, bilimsel hesaplama için önemli bir araçtır ve dünya çapında bilimsel çalışmaların hızla artan bir payını hesaba katar.

Python, hızlı gelişmeye uygun üst düzey bir dildir. Birçok kütüphane tarafından desteklenen nispeten küçük bir çekirdek dile sahiptir. Çoklu programlama stilleri desteklenir (prosedürel, nesne yönelimli, işlevsel vb.) Python derlenmek yerine yorumlanır.

Python görüntü işleme alanı ile ilişkili algoritmaları içerisinde barındıran bilgisayarlı görü kapsamında yapay zekâ çalışmalarında da oldukça yaygın kullanılan bir programlama dili olmuştur. Pythonun, Yapay Zekâ geliştiricileri için aşağıdaki nedenden dolayı çok popüler olduğunu da ifade edebiliriz:

1. Kullanımı kolaydır:

- Python'un kullanımı kolaydır ve hızlı bir öğrenme eğrisine sahiptir. Yeni veri bilimcileri, kullanımı basit sözdizimi ve daha iyi anlaşılabilirliği ile Python'u kolayca öğrenebilir.
- Python ayrıca verilerin daha iyi işlenmesine yardımcı olan birçok veri madenciliği aracı sunar, örneğin Rapid Miner, Weka, Orange vb.
- Python, Pandas, NumPy, SciPy, TensorFlow gibi birçok kullanışlı ve kullanımı kolay kitaplığa ve yetenekli bir Python programcısının iyi bilmesi gereken daha birçok konsepte sahip olduğundan veri bilimcileri için önemlidir.

2. Python esnektir:

- Python yalnızca yazılım oluşturmaya izin vermekle kalmaz, aynı zamanda analiz, sayısal ve mantıksal verilerin hesaplanması ve web geliştirme ile ilgilenmenize de olanak tanır.
- Python ayrıca TurboGears, Django ve Tornado gibi web geliştirme çerçeveleriyle çeşitli önde gelen web sitelerini kontrol ederek web üzerinde her yerde bulunur hale gelmiştir.
- Uygulama ve web geliştirme yeteneğine sahip geliştiriciler için mükemmeldir. Büyük bir sürpriz değil, çoğu veri bilimci, piyasada mevcut olan sonraki programlama alternatifleri için bunu tercih etmektedirler.

3. Python daha iyi analiz araçları oluşturur:

- Veri analitiği, veri biliminin gerekli bir parçasıdır. Veri analitiği araçları, herhangi bir işteki performansı değerlendirmek için önemli olan farklı çerçeveler hakkında bilgi verir. Python programlama dili, veri analizi araçları oluşturmak için en iyi seçimdir.
- Python kolayca daha iyi bilgi sağlayabilir, örnekler alabilir ve büyük veri kümelerinden verileri ilişkilendirebilir. Python, self servis analitikte ayrıca önemlidir. Python da benzer şekilde veri madenciliği organizasyonlarının verileri kendi iyiliği için kullanma olasılıklarını artırmalarına yardımcı oldu.

4. Python, derin öğrenme için önemlidir:

- Python, derin öğrenme algoritmaları geliştirmede veri bilimcilerine yardımcı olan TensorFlow, Keras ve Theano gibi birçok pakete sahiptir. Python, derin öğrenme algoritmaları konusunda üstün yardım sağlar.

- Derin öğrenme algoritmaları, insan beyni mimarisinden esinlenmiştir. İnsan zihninin davranışını yeniden canlandıran yapay sinir ağları kurmayı başarır. Derin öğrenme sinir ağları, farklı girdi parametrelerine ağırlık ve önyargı verir ve istenen çıktıyı verir.

5. Büyük topluluk tabanı:

- Python, mühendislerden ve veri bilimcilerinden oluşan devasa bir topluluk tabanına sahiptir: Python.org, Fullstackpython.com, realpython.com vb. Python geliştiricileri, sorunlarını ve düşüncelerini topluluğa aktarabilir. Python Paket Dizini, Python programlama dilinin farklı ufuklarını keşfetmek için olağanüstü bir yerdir. Python geliştiricileri, zaman içinde daha iyi olmasına yardımcı olan dilde sürekli iyileştirmeler yapmaya devam etmektedirler.

3.3 GÖRÜNTÜ İŞLEMEDE KULLANILAN PYTHON KÜTÜPHANELERİ

Aşağıdaki kütüphaneler, python'da Görüntü işleme sırasında kullanılmaktadırlar;

Scikit-image, SciPy, Mahotas, Simple ITK, Pillow, Matplotlib ve OpenCV

Scikit-image: Aynı NumPy üyeleri tarafından çalıştırılan açık kaynaklı bir Python paketidir (Young ve diğ., 2007). Araştırma, akademik ve endüstriyel kullanım için algoritmalar ve kaynaklar kullanır. Python ekosistemine yeni başlayanlar için bile basit ve anlaşılır bir kütüphanedir. Kod yüksek kalitededir, araştırmacılar tarafından derlenir ve aktif çalışan bir gönüllü topluluğu tarafından sürekli güncellenmektedir. Scikit-Image ile ilgili belgelere <https://scikit-image.org/docs/stable/> web adresinden ulaşılabilir.

SciPy: SciPy, Python'un temel bilim modüllerinden biridir (NumPy gibi) ve temel inceleme ve işleme görevleri için kullanılabilir. Özellikle, scipy.ndimage alt modülü (SciPy v1.1.0) üzerinde çalışan fonksiyonlar sağlar. Bu paket şu anda doğrudan ve çevrimdışı filtreleme işlevlerini, ikili morfolojiyi, B-spline çevirisini ve nesne derecelendirmelerini içerir. SciPy ile ilgili belgelere <https://www.scipy.org/docs.html> web adresinden ulaşılabilir.

Mahotas, başka bir Python bilgisayarı ve grafik editörüdür. Morfolojik filtreleme ve işlevsellik gibi geleneksel görüntü işleme işlevlerinin yanı sıra ilgi çekici noktaların keşfi ve yerel tanımlar dahil olmak üzere modern bilgisayar destekli hesaplamalı hesaplama işlevlerini içerir (Coelho, 2013). Ekran, hızlı geliştirme için uygun olan Python'dadır, ancak algoritmalar C++ 'da kullanılır. Mahotas Kütüphanesi, az kod içeren düşük güvenlikte olan hızlı bir kütüphanedir. Mahotas ile ilgili belgelere <https://mahotas.readthedocs.io/> web adresinden ulaşılabilir.

ITK (Insight Segmentation and Registration Toolkit), geliştiricilere kapsamlı bir dizi görüntü analizi yazılımı araçları ve çevrilmiş diller sağlayan açık kaynaklı bir kısayol sistemidir. Bu sistem aynı zamanda standart filtreleme, görüntü sınıflandırma ve kayıt işlevlerini destekleyen çok sayıda özelliğe sahip bir görüntü analiz aracıdır (Bovik, 2010). Bu başlık altında geçen SimpleITK, C++ ile yazılmıştır ancak Python dahil olmak üzere çok sayıda programlama dilinde mevcuttur. SimpleITK ile ilgili belgelere <https://simpleitk.readthedocs.io/en/master/> web adresinden ulaşılabilir.

PIL (Python Imaging Library), çoklu görüntü dosyası formatlarını açma, yönetme ve saklama desteği ekleyen ücretsiz bir Python programlama dili kütüphanesidir (Russ, 2016). Ancak, 2009'daki son sürümle gelişimi durmuştur. Lakin Pillow, PIL şekilli çatılaşma ile, kurulumu gayet kolay ve tüm büyük işletim sistemlerinde çalışır bir format geliştirilmiştir ve bu format Python 3.x versiyonlarında renk-uzay dönüşümlerini desteklemektedir. Pillow ile ilgili belgelere <https://pillow.readthedocs.io/en/3.1.x/index.html> web adresinden ulaşılabilir.

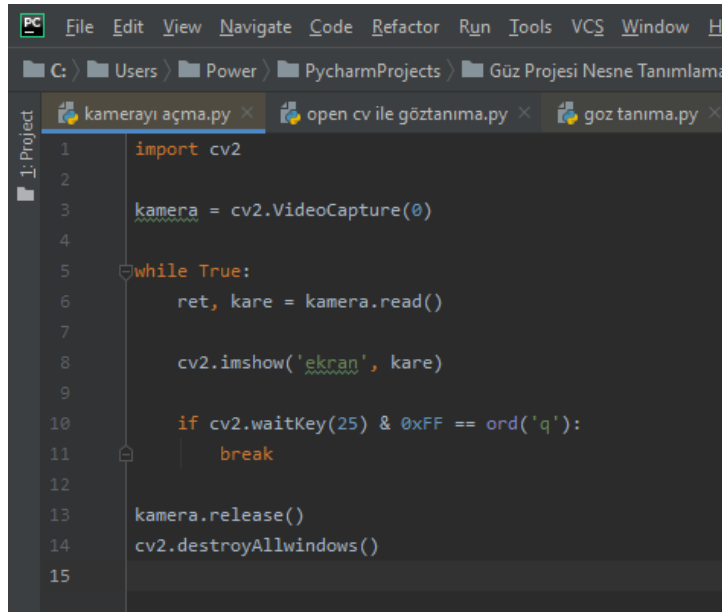
Matplotlib, Python'da statik, hareketli ve etkileşimli görselleştirmeler oluşturmak için kapsamlı bir kitaplıktır. Matplotlib grafiksel olarak zor görünen işlerin kolaylıkla gerçekleştirilebilmesinin mümkün kılacağı (Tosi, 2009). En önemli avantajlarından birisi çok fazla kod yazmadan sadece birkaç satırlık kodlar çeşitli grafikler, histogramlar, güç spektrumları, çubuk grafikler, hata çizelgeleri, saçılım grafiği oluşturulabilmesini sağlamasıdır. Yayın kalitesinde grafikler oluşturmaya fırsat veren, yakınlaştıran, kaydırabilen ve güncellenebilen etkileşimli figürler yapmaya olanak verir. Görsel stili ve düzeni özelleştirmeye de fırsat tanır. Matplotlib ile ilgili belgelere <https://matplotlib.org/> web adresinden ulaşılabilir.

3.4 OPENCV

OpenCV (Open Source Computer Vision = Açık Kaynak Bilgisayarla Görü), açık kaynak kodlu bir bilgisayarlı görü ve makine öğrenmesi yazılım kütüphanesidir. OpenCV, bilgisayarlı görü uygulamaları için ortak bir altyapı sağlamak ve ticari ürünlerde makine öğrenme algısının kullanımını hızlandırmak amacıyla oluşturulmuştur. BSD lisanslı bir ürün olan OpenCV, işletmelerin kodu kullanmasını ve değiştirmesini kolaylaştırır (Internet 2). OpenCV kütüphanesi hem klasik hem de son teknoloji bilgisayarlı görme ve makine öğrenimi algoritmalarından oluşan kapsamlı bir set içeren 2500'den fazla optimize edilmiş algoritmaya sahiptir. Bu algoritmalar, yüzleri algılamak ve tanımak, nesneleri tanımlamak, videolardaki insan eylemlerini sınıflandırmak, kamera hareketlerini izlemek, hareketli nesneleri izlemek, nesnelerin 3B modellerini çıkarmak, stereo kameralardan 3B nokta bulutları üretmek ve görüntüleri bir araya getirerek yüksek bir görüntü elde etmek, bir görüntü veri tabanından benzer görüntüler bulmak, flaş kullanılarak çekilen görüntülerden kırmızı gözleri kaldırmak, göz hareketlerini takip etmek için kullanılabilir. OpenCV, kullanıcı topluluklarında 47 binden fazla kişiye ve 18 milyonu aşan tahmini indirme sayısına sahiptir. Kütüphane, şirketler, araştırma grupları ve devlet kurumları tarafından yaygın olarak kullanılmaktadır. OpenCV kütüphanesi, C++, Python, Java ve MATLAB arayüzlerine sahiptir ve Windows, Linux, Android ve Mac OS destekler. 500'den fazla algoritma ve bu algoritmaları oluşturan veya destekleyen yaklaşık 10 kat fonksiyon barındırmaktadır. OpenCV natif olarak C++ programlama dili ile yazılmıştır ve STL kapsayıcılarıyla sorunsuz çalışan şablonlu bir arayüze sahiptir. Bu yazılım, bir dizi görüntü işleme işlevinin yanı sıra görüntü ve desen analizi işlevleri sağlar. OpenCV'deki bazı uygulama örnekleri, yüzleri algılama ve tanıma, nesneleri tanımlama, videodaki insan eylemlerini sınıflandırma, kamera hareketlerini izleme, hareketli nesneleri izleme gibi Nesne Tanımlama, Bölümleme ve Tanıma'dır. OpenCV, bilgisayarlı görü görüntülerinin işlenmesinde çok iyi bilinen bir kütüphanedir. OpenCV, gerçek zamanlı uygulamalar için tasarlanmıştır ve görüntü/video için iyi işlevlere sahiptir. OpenCV, bilgisayarla görme alanında çalışacak programcılar için şiddetle tavsiye edilir, çünkü bu kütüphane, güvenilir, dijital görme alanında güçlü ve insanlarda görsel işlemeye benzer yeteneklere sahip uygulamalar oluşturabilmektedir (Widodo ve diğ., 2020).

3.5 GÖZ KIRPMA SAYISININ OPENCV KÜTÜPHANESİ KULLANILARAK ÖLÇÜLMESİ

Bu tezde, kuru göz sendromunun gerek patogenezinde gerekse tedaviye cevabın değerlendirilmesinde klinik öneminin çalışmalar ile kanıtlandığı göz kırpma sayısının python programlama dili ile yazılmış kodlar ile görüntü işleme kütüphanelerinden birisi olan OpenCV'nin desteği ile objektif olarak sayılması amaçlanmıştır, bu bağlamda kod satırları oluşturulmuştur. Öncelikli olarak terminale kod satırlarını girmeden önce OpenCV görüntü işleme kütüphanesinin yüklenmesi gerekmektedir. Tezin yapıldığı bilgisayarda yüklü olan Python programını PyCharm kod editörü ekranına 'import cv2' kod satırı girilerek openCV kütüphanesinin yüklenmesi yapılmıştır. Daha sonra OpenCV'de video ve kameradan canlı görüntü alınması için 'kamera=cv2.VideoCapture(x)' kod satırı girilmiştir. Buradaki 'x' değişkeni kamera değişkenine atadığımız görüntü kaynağını belirtmekte olup üç değer (0, 1, dosya_adi.mp4) den birisini alabilmektedir. Burada '0' bilgisayar üzerindeki tanımlı kameradan görüntü alındığını, '1' ise bilgisayara USB aracılığı ile bağlanabilecek harici bir kameradan görüntünün alınacağını ifade etmektedir. 'dosya_adi.mp4' ise belirlenen dizindeki daha önceden kaydedilmiş bir videonun da kaynak olabileceğini işaret etmektedir. Terminale girilen diğer bir satırda ise 'while' döngüsü oluşturulmuştur.



```
1 import cv2
2
3 kamera = cv2.VideoCapture(0)
4
5 while True:
6     ret, kare = kamera.read()
7
8     cv2.imshow('ekran', kare)
9
10    if cv2.waitKey(25) & 0xFF == ord('q'):
11        break
12
13 kamera.release()
14 cv2.destroyAllWindows()
15
```

Şekil 2. Gerçek Zamanlı Video Görüntüsü Alınmasını Sağlayan Python Kodları

Şekil 2’de de görüldüğü üzere; ‘while True’ satırı ile break oluşturuluncaya kadar döngü oluşturulması amaçlanmaktadır. Takip eden satırda bulunan ‘ret,kare = kamera.read()’ kodu ile ret ve kare şeklinde iki değişkenin oluşturulması amaçlanmıştır. ‘ret’ değişkeni kamera açık olduğu sürece true’ya dönecek ‘kare’ değişkeni her 25 ms’de bir fotoğraf çekecektir. Böylece bir videonun oluşturulması amaçlanmıştır. ‘cv2.imshow(‘ekran’, kare)’ satırı ile de ‘ekran’ üst başlığı ile açılan pencerede her 25 ms’de bir numpy sınıfında bir obje BGR kodları ile çağrılmakta ve bunun peşi sıra gösterilmesi ile video oynatılmış oluyor. Bu satırdan sonra while döngüsü içine bir if bloğu da açılıyor. ‘If cv2.waitKey(25) & 0xFF == ord(‘q’)’ kodu ile de 25ms bekleyip daha sonra kapatma işlemi bir bitwise operatörü olan klavyedeki ‘q’ karakteri basılarak ‘ekran’ penceresinden çıkılmayı sağlıyor. While döngüsünden çıkıldıktan sonra ‘kamera.release ()’ ile kameradan görüntü alınması kesilmekte ‘cv2.destroyAllWindows()’ kodu ile de tüm pencerelerin kapatılması sağlanmaktadır.

OpenCV kütüphanemizi yükledikten sonra göz kırpma sayacımızı tanımlayabilmemiz için bu kütüphanenin içinden iki adet .xml dosyasını almamız gerekiyor bu amaçla ‘goz_cascade = cv2.CascadeClassifier(‘haarcascade_eye.xml’)’ ve ‘yuz_cascade = cv2.CascadeClassifier(‘haarcascade_frontalface_default.xml’)’ satır kodlarını giriyoruz. Böylece openCVdeki ilgili xml dosyalarını programa çağırıyoruz. BGR renk kodlarını haarcascade classifier’ın rahat çözebilmesi için gri tonuna çevirme işlemi gerçekleştiriyoruz. Bunun için ‘gri_ton = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)’ satırını yazıyoruz. Böylece görüntü üzerindeki her bir satırdaki pikseli BGR düzleminden GREY düzlemine çeviriyoruz (3 tane renk skalasını tek bir renk skalasına indirgiyoruz). Çünkü haarcascade sınıflandırıcısındaki .xml dosyalarını kullanabilmemiz için renkli resmin gri tona çevrilmesi gerekiyor. Daha sonra ‘gozler = goz_cascade.detectMultiScale (gri_ton, 1.1, 4)’ kod satırı ile gözleri tespit etmek için detectMultiScale fonksiyonunu kullanarak ile gözleri koordinatları belli olacak şekilde liste şeklinde döndürüyoruz. Yine ‘1.1’ değerini girerek de resim boyutunu bir miktar büyütülüp daha fazla piksel aralığı oluşturulup buna göre filtreleme yapıyoruz. En sondaki ‘4’ değişkeni ile de resim 4 kez taranarak daha keskin bir sonuç alıyoruz. Aynı şey ‘yuzler = yuz_cascade.detectMultiScale (gri_ton, 1.3, 5)’ kod satırı ile daha geniş bir alanı tarayan yüz için yapıyor. Daha sonra ‘for (x,y,w,h) in gozler: cv2.rectangle (kare, (x, y), (x+w, y+h), (255, 0, 0), 2)’

kod satırı ile de gözler diye oluşturulan bir listenin içine girilip kare isimli değerde x, y karenin sol üst köşesini, x+w, y+h sağ alt köşesi belirleniyor. Alt satıra da for döngüsü oluşturulup ile birden fazla yüzün bulunduğu gözleri aynı framede gösterebiliyoruz. Benzer 'for (x,y,w,h) in yuzler: cv2.rectangle (img, (x, y), (x+w, y+h), (255, 0, 0), 2)' kodu ile de bu sefer yüzün sınırları belirleniyor. Bu iki alan belirlendikten sonra bunlar sayısal değerlere dönüştürülüyor ve bu iki sayısal değer arası fark da '.... DEFA GÖZ KIPTINIZ' şeklinde sonuç veriyor. Aşağıdaki 3. Şekilde tüm kod satırı ile denek üzerindeki göz kırpma sayısını çıktı olarak veren sonucu görmekteyiz (Şekil 4 ve Şekil 5).

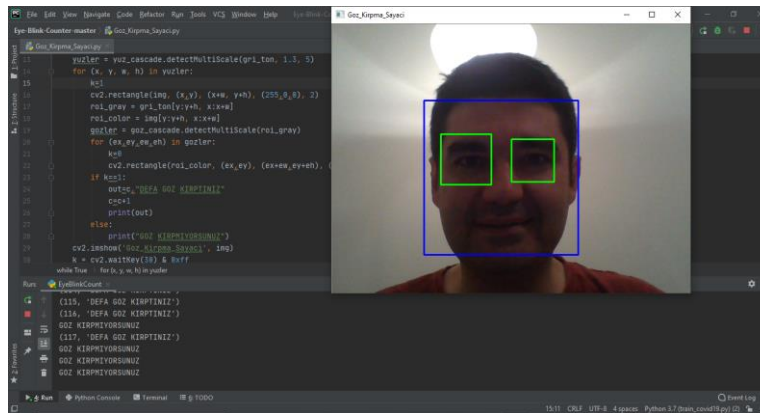
```
import cv2
from cv2 import CascadeClassifier

yuz_cascade = CascadeClassifier('haarcascade_frontalface_default.xml')
goz_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

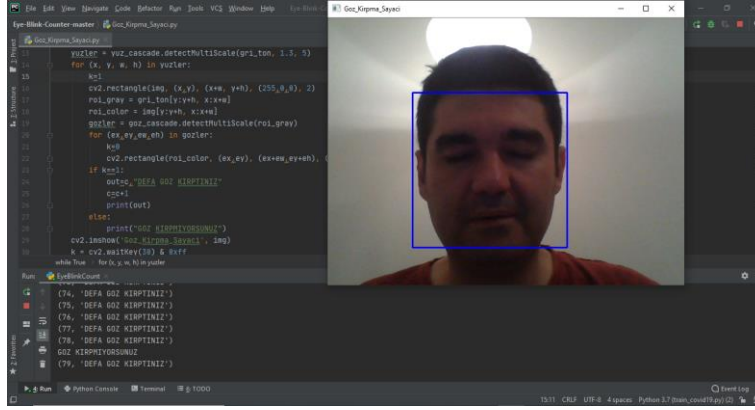
kamera = cv2.VideoCapture(0)
c=0

while True:
    ret,img = kamera.read()
    gri_ton = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    yuzler = yuz_cascade.detectMultiScale(gri_ton, 1.3, 5)
    for (x, y, w, h) in yuzler:
        k=1
        cv2.rectangle(img, (x,y), (x+w, y+h), (255,0,0), 2)
        roi_gray = gri_ton[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        gozler = goz_cascade.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in gozler:
            k=0
            cv2.rectangle(roi_color, (ex,ey), (ex+ew,ey+eh), (0,255,0), 2)
            if k==1:
                out=c,"DEFA GÖZ KIPTINIZ"
                c=c+1
                print(out)
            else:
                print("GÖZ KIRPMIYORSUNUZ")
        cv2.imshow('Göz_Kirpma_Sayaci', img)
        k = cv2.waitKey(30) & 0xff
        if k==27:
            break
kamera.release()
cv2.destroyAllWindows()
```

Şekil 3. Gerçek Zamanlı Videodan Göz Kırpma Sayısını Ölçümünü Sağlayan Python Kodları



Şekil 4. Gerçek Zamanlı Videodan Gözler Açıkken Ortaya Çıkan Göz ve Yüz Konturlarını Tanımlayan Python Kodları



Şekil 5. Gerçek Zamanlı Videoda Gözler Kapalıyken Ortaya Çıkan Yüz Konturları ve Çıktıda Göz Kırpma Sayısını Gösteren Python Kodları

BÖLÜM IV (VERİ)

Bu bölümde, çalışmamızdaki katılımcılar hakkında bilgilere yer verilmektedir.

4.1 HASTA SEÇİMİ

Bu tez için, Acıbadem Üniversitesi Tıp Fakültesi Klinik Araştırmalar Etik Kurulu tarafından 26.05.2021 tarihinde 2021 / 09 protokol numarası ile onay alınmıştır. Çalışmamızda 06.2021 ile 12.2021 tarihleri arasında kuru göz semptomatolojisi ile Atakent Eğitim Araştırma Hastanesi Göz Kliniğine başvuran hastalara konvansiyonel tetkikler (TBUT <10sn, OSDI>12 ve Schirmer testi <10 mm / 5dk) ile kuru göz sendromu tanısı klinik olarak konulmuştur. Bu bulgulardan herhangi bir tanesine sahip olmayan, göz kliniğine rutin gözlük muayenesine gelen, sistemik hastalığı olmayan, herhangi bir göz damlası kullanmayan ve yaş ile cinsiyet olarak kuru göz sendromu tanısı alan hasta grubuna benzer özellikler gösteren bireyler de sağlıklı kontrol grubu olarak çalışmaya dahil edilmişlerdir.

Her iki gruptaki katılımcılara tamamen ücretsiz ve açık kaynak kodlu öğrenilmesi kolay, zengin bir modül (kütüphane) kaynağına sahip python diliyle yazılan kod ile göz kırpma sayısı ölçülmüştür. Bu tez çalışmasında OpenCV kütüphanesinin kullanımı ile oluşturulan kod; normal ve kuru göz semptomları olan hastalar

dikkatlerini dağıtabilecek unsurların ortadan kaldırıldığı sessiz bir ortamda çeşitli aktivitelerde (task 1 ve task 2) bulunmuşlar ve bu farklı dikkat düzeyi gereken görevlerde gerçekleştirdikleri göz kırpmaları ölçülmüştür. Bu görevlerden ‘task 1’ olarak adlandırılan aktivite bir sayfa olan A4 boyutundaki basılı materyali 5 dk’da okuma şeklinde tanımlanmıştır. ‘Task 2’ aktivite ise basılı materyal olarak verilen metnin Word dokümanı olarak dijital ekranda 5 dk’da okunması şeklinde tanımlanmıştır. Ayrıca objektif olarak göz kırpması miktarını ölçen bu kod vasıtasıyla elde edilen göz kırpması sayılarının kuru göz tanısında kullanılan diğer konvansiyonel tanı metodları (OSDI skoru, Schirmer testi değeri ve TBUT değeri) ndaki sayısal değerler ile korelasyonuna da bakılmıştır.

4.2 İSTATİSTİKSEL ANALİZLER

Çalışmadaki verilerin istatistik analizi Statistical Package for Social Sciences (SPSS) (IBM SPSS, ver 28.0, Armonk, NY: IBM Corp., ABD) ile yapıldı.

Verilerin normal dağılım koşulu Shapiro-Wilk testi ile test edilmiştir. Tanımlayıcı istatistikler olarak: Ortalama, standard sapma, minimum ve maksimum değeri ile median ve %25 ile %75 persentil değerleri kullanılmıştır. Normal dağılım gösteren kuru göz grubu ile normal grup arasındaki istatistiksel farklılığa bağımsız gruplarda t- testi ile, aynı grup hasta içindeki farklı aktiviteler arasındaki istatistiksel farklılığa ise bağımlı gruplarda t testi kullanılarak araştırılmıştır. Aynı grup içindeki sağ ve sol göz arasındaki konvansiyonel test sonuçları ve farklı aktivitelerde ölçülen göz kırpması sayıları yine bağımlı gruplarda t-testi ile karşılaştırılmıştır. İki faktörle tekrarlanan ölçümlerde birinci aktivite ile ikinci aktivite arasındaki fark ayrı ayrı ölçülmüş ve gruplar arasındaki farkın karşılaştırılması da yapılmıştır. Gerek kuru göz gerekse normal grup içerisinde konvansiyonel testlerin sonuçları ile göz kırpması sayısı arasındaki korelasyon Pearson korelasyon analizi ile değerlendirilmiştir. Göz kırpması sayısı ölçümleri sürekli olduğu için, gözlemci içi uyumun değerlendirilmesinde sınıf içi korelasyon katsayısı hesaplanmıştır. Konvansiyonel tetkiklerin her birisi ile göz kırpması sayısı arasında linear regresyon eşitlikleri de yine bu tez kapsamında hesaplanmıştır.

BÖLÜM V (BULGULAR)

Bu bölümde kullanılan yöntemler ile yapılan deneylerden elde edilen sonuçlar yer almaktadır.

5.1 TANIMLAYICI İSTATİSTİKLER ve GRUPLAR ARASINDAKİ FARKLAR

Kuru göz hastalığı olan 36 hasta ile bu gruba yaş ve cinsiyet olarak benzer özellik gösteren sağlıklı gruptaki 36 hastanın verilerinin dağılımlarına bakıldığında kuru göz grubunda Schirmer testi, TBUT değeri, OSDI skoru ve Python kodu ile task 1 ve task 2 değerlerinin p değerlerinin >0.05 olduğu gözlenmektedir, dolayısı ile kuru göz grubunda veriler normal dağılım göstermektedir. Lakin normal gruba bakıldığında Python kodu ile ölçülen göz kırpma sayısının task 1 de sadece $p > 0.05$ değerinde çıktığı ve bu datanın dağılımının normal olduğu, diğer tüm değişkenlerin normal grupta normal dağılım göstermediği saptanmıştır. Bu durumda da nonparametrik testler gruplararası istatistiksel farklılık araştırılmıştır. Her iki gruptaki tüm değişkenler için p değerleri Tablo 1’de özetlenmiştir.

Tablo 1. Kuru Göz Sendromu ve Normal Gruplardaki Dağılımın Tespiti için Gerçekleştirilen Normalite Testi Sonuçları

Normalite Testleri		
KURU GÖZ / NORMAL		Shapiro-Wilk
		Sig.
Kuru Göz	Schirmer Testi_Sağ	,578
	Schirmer Testi_Sol	,151
	TBUT_Sağ	,396
	TBUT_Sol	,066

	OSDI	,421
	PYTHON KODU İLE TASK 1 GÖZ KIRPMA SAYISI	,477
	PYTHON KODU İLE TASK 2 GÖZ KIRPMA SAYISI	,608
Normal	Schirmer Testi_Sağ	,039
	Schirmer Testi_Sol	,019
	TBUT_Sağ	,026
	TBUT_Sol	,022
	OSDI	,009
	PYTHON KODU İLE TASK 1 GÖZ KIRPMA SAYISI	,344
	PYTHON KODU İLE TASK 2 GÖZ KIRPMA SAYISI	,092

Kuru göz sendromlu hasta grubundaki 36 hastanın ortalama yaşı 43.28 ± 12.74 y (19-76 y) iken normal gruptaki katılımcıların ortalama yaşının 42.53 ± 11.47 y (18-66 y) olduğu gözlenmiştir. Bu fark istatistiksel olarak anlamlı bulunmamıştır ($p=0.794$, bağımsız gruplarda t-test). Kuru göz sendromlu hasta grubunda konvansiyonel testlerden Schirmer testi değerinin sağ ve sol gözde sırasıyla 7.28 ± 2.42 mm ve 7.17 ± 2.40 mm düzeylerinde olduğu, normal grupta yine aynı testin değerinin sağ ve sol gözde sırasıyla 30.19 ± 3.74 mm ve 30.31 ± 3.58 mm düzeylerinde olduğu saptanmış olup Schirmer testi değerlerinin normal grupta istatistiksel olarak anlamlı yüksek olduğu gözlenmiştir ($p<0.05$, bağımsız gruplarda t-test). Bir başka konvansiyonel test olan TBUT değerlerinin de kuru göz sendromlu gruptaki hastalarda sağ ve sol gözde sırasıyla 9.83 ± 3.29 sn ve 10.03 ± 3.38 sn düzeylerinde olduğu, normal grupta ise bu değerlerin sağ ve sol gözde sırasıyla 36.28 ± 6.63 sn ve 35.97 ± 6.71 sn düzeylerinde olduğu saptanmış ve bu değerlerin de gruplar arasında istatistiksel olarak anlamlı farklı olduğu not edilmiştir. Subjektif şikayetlerin bir göstergesi olan OSDI skorlarının ise kuru göz sendromlu gruptaki hastalarda 35.28 ± 7.55 , normal grupta ise 5.25 ± 2.42 düzeyinde olduğu saptanmıştır. ($p<0.05$, bağımsız gruplarda t-test). Python kodu ile ölçülen ortalama göz kırpma sayısının task 1 aktivitede kuru göz ile normal gruplarda sırasıyla 25.97 ± 4.40 ve 12 ± 2.63 ($p<0.05$) olduğu saptanmıştır. Task 2 aktivitede ise

ölçülen ortalama göz kırpma sayısının kuru göz ile normal gruplarda sırasıyla 20.56 ± 4.56 ve 8.92 ± 2.29 ($p < 0.05$) olduğu saptanmıştır. Tüm değerler tablo 2’de özetlenmiştir.

Tablo 2. Kuru Göz Sendromu ve Normal Bireylerdeki Tanımlayıcı İstatistik Sonuçları

		N					Persentiller		
	KURU GÖZ / NORMAL		Ort	SS	Min	Maks	25th	Ortanca	75th
YAŞ	Kuru Göz	36	43,28	12,74	19,00	76,00	33,25	43,00	50,75
	Normal	36	42,53	11,47	18,00	66,00	33,50	43,00	50,00
Schirmer Testi_Sağ	Kuru Göz	36	7,28	2,42	2,00	12,00	6,00	7,00	9,00
	Normal	36	30,19	3,74	22,00	35,00	27,25	30,00	34,00
Schirmer Testi_Sol	Kuru Göz	36	7,17	2,40	3,00	12,00	5,25	7,00	9,00
	Normal	36	30,31	3,58	24,00	35,00	28,00	30,00	34,00
TBUT_Sağ	Kuru Göz	36	9,83	3,29	4,00	17,00	7,00	9,50	12,00
	Normal	36	36,28	6,63	26,00	50,00	30,25	35,00	41,50
TBUT_Sol	Kuru Göz	36	10,03	3,38	4,00	16,00	7,00	9,00	12,75
	Normal	36	35,97	6,71	24,00	52,00	31,25	34,50	40,00
OSDI	Kuru Göz	36	35,28	7,55	23,00	55,00	28,50	36,00	40,00
	Normal	36	5,25	2,42	1,00	12,00	4,00	5,00	6,00
PYTHON KODU İLE TASK 1 GÖZ KIRPMA SAYI	Kuru Göz	36	25,97	4,40	18,00	36,00	22,25	26,00	29,75
	Normal	36	12,00	2,63	7,00	17,00	10,00	12,00	14,00
PYTHON KODU İLE TASK 2 GÖZ KIRPMA SAYI	Kuru Göz	36	20,56	4,56	12,00	31,00	17,00	21,00	24,00
	Normal	36	8,92	2,29	5,00	15,00	7,00	9,00	10,00

5.2 KURU GÖZ GRUBU VE SAĞLIKLI GRUPLARDA FARKLI TASKLARDAKİ GÖZ KIRPMA SAYILARININ KARŞILAŞTIRILMASI

Kuru göz grubunda python kodu ile ölçülen göz kırpma sayısı ortalaması task 1’de 5.41 kez daha fazla saptanmış olup bu fark istatistiksel olarak anlamlı bulunmuştur ($p<0.01$, paired samples test).

Normal grupta da python kodu ile ölçülen göz kırpma sayısı ortalaması task 1’de 3.08 kez daha fazla saptanmış olup bu değer de istatistiksel olarak anlamlı bulunmuştur ($p<0.01$, paired samples test).

Ayrıca hem kuru göz hem de normal grupta her iki göz arasındaki schirmer testi ve TBUT testleri sonuçları istatistiksel olarak anlamlı farklılık göstermemiştir (kuru göz grubunda Schirmer Sağ / Sol farkı için $p=0.53$, TBUT Sağ / Sol farkı için $p=0.14$; normal göz grubunda Schirmer Sağ / Sol farkı için $p=0.51$, TBUT Sağ / Sol farkı için $p=0.28$;). Sonuçlar aşağıdaki tablo 3’de özetlenmiştir

Tablo 3. Kuru Göz Sendromu ve Normal Bireylerdeki Farklı Tasklardaki Göz Kırpma Sayısının Karşılaştırılması

					95% Farkın Güven Aralığı				
		Ort.	Std. Sapma	Std. Hata	Alt	Üst			Sig (p)
Kuru Göz	PYTHON KODU İLE TASK 1 SAYI - PYTHON KODU İLE TASK 2 SAYI	5,417	1,204	,201	5,009	5,824	26,990	35	,000
Normal	PYTHON KODU İLE TASK 1 SAYI - PYTHON KODU İLE TASK 2 SAYI	3,083	1,025	,171	2,737	3,430	18,054	35	,000

5.3 KONVASİYONEL TESTLER İLE GÖZ KIRPMA SAYISI DEĞERLERİNİN KORELASYONU

Sınıf içi korelasyon katsayısı (ICC) değeri hem Schirmer testi (Tablo 4) hem de TBUT (Şekil 5) için ortalama ölçümlerinde 0.998 düzeyinde bulunmuştur.

Tablo 4. Schirmer testi için Sınıf içi Korelasyon Katsayısı (ICC)

	Sınıf içi Korelasyon	95% Güven Aralığı		F Test (Gerçek Değer 0 ile)			
		Alt Sınır	Üst Sınır	Değer	df1	df2	Sig (p)
Tekli Ölçüm	,996	,994	,998	544,296	71	72	,000
Ortalama Ölçüm	,998	,997	,999	544,296	71	72	,000

Tablo 5. TBUT için Sınıf içi Korelasyon Katsayısı (ICC)

	Sınıf içi Korelasyon	95% Güven Aralığı		F Test (Gerçek Değer 0 ile)			
		Alt Sınır	Üst Sınır	Değer	df1	df2	Sig (p)
Tekli Ölçüm	,996	,993	,997	466,706	71	72	,000
Ortalama Ölçüm	,998	,997	,999	466,706	71	72	,000

Tablo 6. Python Kodu ile Ölçülen Göz Kırpma Sayısı için Sınıf içi Korelasyon Katsayısı (ICC)

	Sınıf içi Korelasyon	95% Güven Aralığı		F Test (Gerçek Değer 0 ile)			
		Alt Sınır	Üst Sınır	Değer	df1	df2	Sig (p)
Tekli Ölçüm	,826	,736	,887	10,496	71	72	,000
Ortalama Ölçüm	,905	,848	,940	10,496	71	72	,000

Sınıf içi korelasyon katsayısı (ICC) python kodu ile ölçülen göz kırpma sayısı için ise 0.905 düzeyinde bulunmuştur (Tablo 6).

Korelasyonlara bakıldığında ise; kuru göz grubunda python kodu ile yazılmış kod ile task 1 aktivite sırasında ölçülen göz kırpma sayısının Schirmer testi ($r = -0.92, p < 0.01$), TBUT test ($r = -0.89, p < 0.01$) ve OSDI skoru ($r = 0.82, p < 0.01$) için güçlü düzeyde korelasyon gösterdiği saptanmıştır.

Normal grupta ise python kodu ile yazılmış kod ile task 1 aktivite sırasında ölçülen göz kırpma sayısının Schirmer testi ($r = -0.92, p < 0.01$), TBUT test ($r = -0.87, p < 0.01$) ve OSDI skoru ($r = 0.68, p < 0.01$) ile güçlü düzeyde korelasyon gösterdiği saptanmıştır.

Kuru göz grubunda python kodu ile yazılmış kod ile task 2 aktivite sırasında ölçülen göz kırpma sayısının schirmer testi ($r = -0.86, p < 0.01$), TBUT test ($r = -0.87, p < 0.01$) ve OSDI skoru ($r = 0.82, p < 0.01$) ile güçlü düzeyde korelasyon gösterdiği saptanmıştır.

Normal grupta ise python kodu ile yazılmış kod ile task 2 aktivite sırasında ölçülen göz kırpma sayısının schirmer testi ($r = -0.87, p < 0.01$), TBUT test ($r = -0.84, p < 0.01$) ve OSDI skoru ($r = 0.66, p < 0.01$) güçlü düzeyde korelasyon gösterdiği saptanmıştır. Sonuçlar aşağıdaki tablo 7’de özetlenmiştir.

Tablo 7. Python Kodu ile Ölçülen Göz Kırpma Sayılarının Konvansiyonel Testler ile Ölçülen Değerler ile Korelasyonun Değerlendirilmesi

KURU GÖZ / NORMAL		PYTHON KODU İLE TASK 1 SAYI			PYTHON KODU İLE TASK 2 SAYI		
		r	P	N	r	P	N
Kuru Göz	SC_Ort	-,916	,000	36	-,859	,000	36
	TBUT_Ort	-,895	,000	36	-,867	,000	36
	OSDI	,822	,000	36	,815	,000	36
Normal	SC_Ort	-,920	,000	36	-,865	,000	36
	TBUT_Ort	-,868	,000	36	-,842	,000	36
	OSDI	,678	,000	36	,655	,000	36

5.4 KONVASİYONEL TESTLER İLE GÖZ KIRPMA SAYISININ ÖNGÖRÜLEBİLİRLİĞİ- REGRESYON EŞİTLİKLERİ-

Python kodu ile gerek task 1 gerekse task 2 aktivite ile ölçülen göz kırpma sayısının tahmininde konvansiyonel tanı tetkikleri kullanılarak tahmin edilmesine dayalı oluşturulacak regresyon modelinde tüm konvansiyonel tanı parametreleri göz kırpma miktarının %85'den fazlasını açıklamaktadır. Her bir parametre için hesaplanan regresyon eşitlikleri tablo 8'de gösterilmiştir.

Tablo 8. Konvansiyonel Testler ile Göz Kırpma Sayacı Test Sonuçlarının Öngörülebilirliğini İlişkilendiren Regresyon Eşitlikleri

Tanısal Parameter	R ²	$y = a \times \ln(x) + b$	P
Python Kodu ile Task 1 aktivite sırasında ölçülen göz kırpma miktarı			
Schirmer Testi	0.914	$y = 46.274 \times \ln(x) - 1.45$	<0.001
TBUT Testi	0.907	$y = 55.460 \times \ln(x) - 1.71$	<0.001
OSDI	0.917	$y = -16.82 \times \ln(x) + 1.95$	<0.001
Python Kodu ile Task 2 aktivite sırasında ölçülen göz kırpma miktarı			
Schirmer Testi	0.853	$y = 42.48 \times \ln(x) - 1.61$	<0.001
TBUT Testi	0.855	$y = 51.15 \times \ln(x) - 1.91$	<0.001
OSDI	0.887	$y = -12.29 \times \ln(x) + 2.21$	<0.001

Tablo 8’de de görüldüğü üzere konvansiyonel tetkiklerinin her birinin prediktör olarak ayrı ayrı kullanılmasıyla bu prediktörlerin gerek basılı metin gerekse dijital ekranda göz kırpması üzerine bağımsız etkileri de lineer regresyon modeli kullanılarak incelenmiştir.

Yukarıda görülen regresyon denkleminde de belirtildiği üzere basılı materyal okuma sırasında ölçülen göz kırpması miktarı tahmininin isabet derecesi 1.0 e daha yakın olup dijital ekrandan okuma sırasında ölçülen göz kırpması miktarı tahminini isabet derecesinden daha yüksek oranda korelasyon katsayısı ile sonuçlanmıştır. Lakin iki durumda da ölçülen göz kırpması miktarının konvansiyonel testler kullanılarak elde edilecek tahminlerde de isabet derecesinin yüksek olduğunu bu tabloya bakarak ifade edebiliriz. İlgi çeken bir diğer unsur ise hem basılı materyal okuma sırasında ölçülen objektif göz kırpması sayısı üzerine hem de dijital ekrandan okuma sırasında ölçülen objektif göz kırpması sayısı üzerine konvansiyonel tetkiklerden subjektif hasta şikayetleri sonucu oluşan OSDI skorunun diğer konvansiyonel tetkiklere göre daha yüksek regresyon katsayısına sahip olmasıdır. Aslında bu durumu subjektif olarak hasta şikayetleri neticesinde oluşan OSDI skorunun objektif olarak elde ettiğimiz göz kırpması sayısını tahmin etmede güçlü düzeyde bir ilişkide olduğu sonucuna varabiliriz.

BÖLÜM VI (TARTIŞMA)

Bölüm 6’da yapılan tez çalışmasının sonucu değerlendirmesi sunulmuştur.

6. TARTIŞMA

Bu tezde kuru göz sendromu tanısı konvansiyonel tetkikler ile konan hastaların göz kırpması Python programlama dili kullanılarak oluşturulmuş kod vasıtasıyla sayılmış ve sağlıklı bireylerden istatistiksel olarak anlamlı yüksek göz kırpması elde edilmiştir. Bu farklılık gerek basılı materyal (task 1) okuma gerekse dijital ekran (task 2) dan okuma sırasında gözlemlenmiştir. Yine her iki grupta da basılı materyalden okuma sırasında sayılan göz kırpması miktarı dijital ekran okuma sırasında elde edilen göz kırpması sayısından istatistiksel olarak anlamlı yüksek bulunmuştur. Ayrıca objektif olarak göz kırpması miktarını ölçen bu kod vasıtasıyla elde edilen değerlerin kuru göz tanısında kullanılan diğer konvansiyonel tanı metodlarındaki

sayısal değerler ile de yüksek düzeylerde istatistiksel olarak anlamlı olacak şekilde korelasyon gösterdiği de saptanmıştır.

İnsandaki göz kırpması mekanizması, dış zararlı uyaranlara karşı oküler yüzeyi korur ve gözyaşının göz yüzeyine eşit şekilde dağılmasını sağlar. Göz kırpması, oküler yüzey neminin korunmasına, gözyaşlarının drenajına, meibomian bezlerden lipidlerin salgılanmasına ve gözyaşı lipidlerinin preorneal gözyaşı filmi boyunca yayılmasına katkıda bulunarak oküler yüzey bütünlüğünün korunmasında önemli bir rol oynar (Korb ve diğ., 1994). Gözyaşı filminin birincil işlevi, göz kırpması aralıkları sırasında konfor ve kaliteli görüşü korumaktır. Gözyaşı disfonksiyonu, gözyaşı dengesizliği ve tahriş semptomları ile bulanık ve dalgalı görme ile sonuçlanır. Bazı hastalar ayrıca göz kırpması sıklığında bir artış olduğunu da ifade etmektedirler (Girard ve Levy, 2019). Önceleri okuma veya video görüntüleme (VDTS) sırasında göz kırpması sayısının azaldığı bildirilmiş olmasına rağmen (Tsubota ve Nakamori, 1993), bazı gözyaşı disfonksiyonu hastaları bu aktiviteler sırasında daha fazla göz kırpması bildirmektedirler (Kato ve diğ., 2019). Bu güncel tez çalışmasında gözyaşı işlev bozukluğu olan gözlerde göz kırpması hızının arttığı bildirilse de bu artışa video görüntüleme cihazlarının etkisinin de araştırılmış olması bu tez çalışmasının özgünlüğünü ortaya koymuştur. Bulgularımız, stabil gözyaşı filmi olan gözlerde göz kırpması sayısının normal olarak baskılandığını, ancak bu baskılamanın gözyaşı disfonksiyonu / gözyaşı instabilitesi olan kuru göz grubunda kaybolduğunu ve günün sonunda refleks olarak artış olduğunu düşündürmektedir. Gözyaşı disfonksiyonu olan hastalarda artan göz kırpmasının fonksiyon ve yaşam kalitesi üzerindeki etkisinden sık sık söz edilmez ve hafife alınabilir. Ancak biz çalışmamızda kuru göz grubunda elde ettiğimiz OSDI skorlarının göz kırpması sayısı ile yüksek korelasyon gösterdiğini de ortaya koymuş bulunmaktayız. Gözyaşı işlev bozukluğu olan gözlerde video izlerken daha yüksek göz kırpması hızı, gözyaşı dengesizliği veya tahriş semptomları nedeniyle kaliteli görme veya rahatlığı korumak için telafi edici olabilir. Günümüz dünyasında, elektronik ekranların izlenmesi evde, işte, boş zamanlarında ve hareket halindeyken günlük yaşamın büyük bir parçası haline gelmiştir. Masaüstü, dizüstü ve tablet bilgisayarların, akıllı telefonların ve elektronik okuma cihazlarının kullanımı her yerde bulunur hale gelmiştir (Rosenfeld ve diğ., 2012). Gerçekten de yakın tarihli bir rapor, yetişkinlerin günde ortalama olarak yaklaşık 8,5 saatini elektronik ekranları izleyerek geçirdiğini

bildirmektedir (Internet 3). Buradan yola çıkarak literatürdeki birkaç araştırma, bilgisayarla çalışma sırasında göz kırpma hızının düştüğünü bildirmektedir (Tsubota ve Nakamori, 1993; Patel ve diğ., 1991; Schlote ve diğ., 2004). Örneğin, Tsubota ve Nakamori 104 ofis çalışanınin rahat olduklarında, kitap okurken veya elektronik ekranda metin izlerken göz kırpma oranlarını karşılaştırmışlardır. Ortalama göz kırpma oranları rahatken dakikada 22 iken, kitap veya ekranı görüntülerken sırasıyla sadece 10 ve 7 olarak saptanmıştır. Bununla birlikte, bu üç test koşulu, yalnızca sunum yönteminde değil, aynı zamanda görev talebinde de değişiklik göstermiştir. Kontrast azaldıkça (Gowrisankaran ve diğ., 2007) veya görevin bilişsel yetkinlik gerektiren talebi arttıkça göz kırpma hızının azaldığı kaydedilmiştir (Himebaugh ve diğ., 2009; Jansen ve diğ., 2010). Bu nedenlerden dolayı, Tsubota ve Nakamori tarafından gözlemlenen farklılıklar, basılı materyalden elektronik gösterime geçişin bir sonucu olmaktan çok, görev zorluğundaki değişikliklerle ilgili olabilir. Çalışmamızdaki basılı ve dijital ekrandaki göz kırpma sayılarındaki fark bu rasyonelle açıklanabilir. Burada sorulması gereken anahtar soru, bilgisayar kullanımı sırasında göz kırpma düzenindeki değişikliğin özellikle elektronik ekranın görüntülenmesiyle mi ilgili olduğu, yoksa yalnızca zorlu bir yakın görme görevinin gerçekleştirilmesinin sonucu mu olduğudur. Nitekim Doughty (Doughty, 2001), bilgisayar kullanımı sırasında göz kırpma etkinliğini inceleyen çalışmaların hem görevin görsel doğasını hem de kullanıcının psikolojik durumunu dikkate alması gerektiğini belirtmiştir. Bilgisayarın çalışması sırasında göz kırpma hızının azaldığını belirten önceki araştırmalar, deneklerin aynı görevi basılı (elektronik olmayan) biçimde aynı görüntüleme koşulları altında gerçekleştirdikleri eşdeğer bir basılı kopya kontrol denemesini içermemiştir. Elektronik ekranın basılı materyallerle karşılaştırıldığında daha düşük görüntü kalitesinin göz kırpma hızındaki değişimden sorumlu olabileceği öne sürülmüştür (Chu ve diğ., 2011). Nitekim bizim tez çalışmamızda da basılı materyal ile dijital ekran okumaları arasında göz kırpma farklılıkları bu durumu destekler niteliktedir.

Göz kırpma sayısının değerlendirilmesinde programcılık alanı ile yeni düzeyde çalışmalar yapılmaya başlanmış ve göz kırpmasının programlama temelli objektif sayımı üzerine çok sınırlı da olsa yayınlar çıkmaya başlamıştır. Bu kapsamda Won ve ark. ikili görüntülere dayalı bir yanıp sönen algılama algoritması tanımlamıştır (Won ve diğ., 2010). Binarizasyon eşiğinin kritik olduğu ve görüntünün aydınlatma

koşullarına bağlı olduğu; yani, eşiğin otomatik olarak belirlendiği bir önceki normalleştirme sürecinin gerekliliğini gündeme getirmiştir. Won, gözün açık mı yoksa kapalı mı olduğunu tespit etmek için iki özellik kullanmıştır: ilk olarak, açık olduğundan beri biriken siyah piksellerin sayısını belirlemek için ardışık kareleri karşılaştırmıştır. Gözün bakış pozisyonuna göre, iris/gözbebeğinin varlığı daha fazla sayıda siyah piksele yol açmıştır; ikinci olarak, iris yüksekliği ve genişliği arasındaki ilişki ölçülmüştür. Bu iki faktör, çerçeveleri ve dolayısıyla göz kapaklarının kapalı olduğu süreyi belirlemek için bir destek vektör makinesi kullanılarak birleştirilmiştir. Bir başka çalışmada ise, Jiang ve ark. bir göz kırpmasının başlangıcını ve sonunu tespit edebilmişlerdir. Ardışık iki çerçeve arasındaki fark ikili hale getirilmiş ve irisin varlığını belirlemek için morfolojik işlemler kullanılmıştır. İrisin tespiti, önceden optimize edilmesi gereken birkaç eşiğin tanımını gerektiren boyut parametrelerine dayanmaktadır. Bu eşikler için optimal değerlerle, yazarlar tekniklerini kullanarak %99,7 doğruluk bildirmişlerdir (Jiang ve diğ. 2013). Başka bir çalışmada ise Tan ve Zhang tarafından, model tanıma yoluyla iris tespiti için önerilen yöntemlerinde %66'lık bir doğruluk düzeyi rapor edilmiştir (Tian ve diğ., 2000); bu, daha sonra irisin maksimuma göre gerçek konumundan kaynaklanan farklı konfigürasyonlarının hesaba katılması ile daha da geliştirilmiş, bu düzeltmeler ile, yazarlar göz kırpmasının algılanmasında %88'lik bir doğruluk bildirmişlerdir (Tan ve Zhang, 2006). Son olarak, Mitelman ve ark. yazarların ilgili parlaklık ve görüntünün frekans dağılımını inceleyerek açık ve kapalı göz koşulları arasındaki farkları tespit edebildikleri yarı otomatik göz durumu algılama algoritmasını geliştirmişlerdir (Mitelman ve diğ., 2009). Birkaç eşik tanımlamak için bir eğitim süreci gerektiren bu yöntem, iris ve göz bebeği bölgelerinden kaynaklanan parlaklık tepe noktalarına dayanmaktadır. Daha sonra, Bernard ve ark., göz kapaklarının kenarlarına karşılık gelen iki çizgiyi tespit etmek için doğrusal bir görüntü işleme analizi uygulamış ve bunun üzerine, göz kırpmalarını belirlemek için bu iki çizgi arasındaki mesafeyi takip etmişlerdir., böylece bu mesafenin kaybolduğu durumu göz kırpmasının gerçekleştiği şeklinde etiketlemişlerdir (Bernard ve diğ., 2013). Biz çalışmamızda BSD lisansı ile ücretsiz olarak kullanabilen OpenCV kütüphanesini kullanarak programlama bilgisini klinik tecrübemize angaje ederek kütüphane içindeki haarcascade sınıflandırıcılarını

belirleyip gerek zamanlı videodan eř zamanlı olarak gz kırpma sayısını lebildik, bu zellięi nedeniyle alıřmamız zgn nitelik kazanmıřtır.

Bu tezde bazı kısıtlılıklar da izlenmiřtir. ncelikle tasklar iki eřit olarak tanımlandı, daha farklı taskların da tanımlanması farklı gz kırpma sayısı ile sonulanabilecektir. Dięer bir kısıtlılık da katılımcıların gz kırpma sayılarını biliyor olmaları sonuların literatrden farklı olarak fazla yksek veya fazla dřk olmasını aıklıyor olabilir. Dięer bir kısıtlılık ise kuru gz hasta grubunun alt kırımları (meibomian bez hastalıęı gibi)nda gz kırpma sayısı farklı ıkabilmektedir. Bir dięer unsur ise her ne kadar testler aynı klinik odasında ve klinik alıřma saatleri sırasında yapılıyor olsa da testin yapıldıęı sıradaki ortamın nem miktarı gz kırpma miktarını etkileyebilmektedir ki durum da grup iindeki standart sapmanın farklılıęını aıklayabilmektedir.

Bu tezde elde ettięimiz sonular zetlenecek olursa; kuru gz sendromu tanısı konan hastaların gz kırpma sayısı Python programlama dili kullanılarak oluřturulmuř kodlar vasıtasıyla llmř ve bu deęerlerin saęlıklı bireylerden daha yksek olduęu saptanmıřtır. Basılı materyalden okuma sırasında llen gz kırpma miktarı hem kuru gz sendromu hem de saęlıklı grupta anlamlı olarak yksek saptanmıřtır. Ayrıca bu alıřma ile objektif olarak gz kırpma miktarını len bu kod vasıtasıyla elde edilen deęerlerin kuru gz tanısında kullanılan dięer konvansiyonel tanı metodlarındaki sayısal deęerler ile yksek dzeyde korelasyon gsterdięi de saptanmıřtır.

İleride gerekleřtirilecek daha fazla ve daha eřitli task ieren, randomize tek kr alıřma dzenekleri ile mobil uygulamalarda geliřtirmeyi planladıęımız gz kırpma sayısı programı ile daha da objektif alıřmaların gerekleřtirilebileceęi dřnlmektedir. Bu tezdten yola ıkarak zerinde alıřmalarına devam ettięimiz bir uygulamanın ticarileřme potansiyeli de bulunmaktadır. Hatta bu mobil uygulama ile geniř saha alıřmalarının yapılabileceęi ve alıřma kohortunun bylelikle geniřletilerek objektif bir yazılım ile daha da destekleyici nedensellik sonuların elde edilmesi dřnlmektedir.

KAYNAKÇA

Begley C et al, “The relationship between habitual patient-reported symptoms and clinical signs among patients with dry eye of varying severity,” *Invest Ophthalmol Vis Sci.* 44(11):4753-4761 (2003).

Bernard F et al, “Eyelid contour detection and tracking for startle research related eye-blink measurements from high-speed video records,” *Comput. Methods Programs Biomed.* 112, 22–37 (2013).

Bovik AC. ‘Handbook of image and video processing’ Academic press (2010).

Bron AJ et al, “Functional aspects of the tear film lipid layer,” *Exp Eye Res.* 78(3):347-60 (2004).

Cardona G et al, “Blink rate, blink amplitude, and tear film integrity during dynamic visual display terminal tasks,” *Curr Eye Res.* 36(3):190-197 (2011).

Chellappa R, ‘Digital Image Processing’, 2nd Edition, IEEE Computer Society Press, Los Alamitos, CA USA, (1992).

Chu C et al, “A comparison of symptoms after viewing text on a computer screen and hardcopy,” *Ophthalmic Physiol Opt.* 31:29-32 (2011).

Coelho LP, “Mahotas: Open source software for scriptable computer vision,” *Journal of Open Research Software* 1(1), p.e3 (2013).

Craig JP et al, “TFOS DEWS II Definition and Classification Report,” *Ocul Surf.* 15(3):276-283 (2017).

de Monchy I et al, “Combination of the Schirmer I and phenol red thread tests as a rescue strategy for diagnosis of ocular dryness associated with Sjogren's syndrome,” *Invest Ophthalmol Vis Sci* 52:5167-73 (2011).

DEWS Group, “The definition and classification of dry eye disease: report of the definition and classification subcommittee of the international dry eye workshop,” *Ocul Surf* 5(2):75-92 (2007).

Doane MG, “Blinking and the mechanics of the lacrimal drainage system”. *Ophthalmology* 88(8):844-51 (1981).

Doughty MJ, “Consideration of three types of spontaneous eyeblink activity in normal humans: during reading and video display terminal use, in primary gaze, and while in conversation,” *Optom Vis Sci.* 78(10):712-25 (2001).

Gimel'farb, G., Patrice, D., *Image Processing and Analysis: A Primer*, World Scientific Europe Ltd, London UK (2018).

Girard BC, Lévy P, “Dry eye syndrome in benign essential blepharospasm,” *J Fr Ophtalmol* 42:1062–1067 (2019).

Gonzalez R C & Woods RE, ‘*Digital Image Processing*’, 4th Edition, Pearson Education, (2017).

Gowrisankaran S et al, “Eyelid squint response to asthenopia-inducing conditions,” *Optom Vis Sci.* 84:611-9 (2007).

Hanyuda A et al, “Physical inactivity, prolonged sedentary behaviors, and use of visual display terminals as potential risk factors for dry eye disease: JPHC-NEXT study,” *Ocul Surf.* 18:56–63 (2020).

Himebaugh NL et al, “Blinking and tear break-up during four visual tasks,” *Optom Vis Sci.* 86:106-14 (2009).

Internet1 https://tr.wikipedia.org/wiki/Python_Yaz%C4%B1l%C4%B1m_Vakf%C4%B1

Internet 2 <https://github.com/mesutpiskin/computer-vision-guide/blob/master/docs/1-opencv-nedir.md>

Internet 3 <http://www.bls.gov/news.release/ciuaw.nr0.htm>.

Jaiswal S et al, “Ocular and visual discomfort associated with smartphones, tablets and computers: what we do and do not know,” *Clin Exp Optom.* 102(5):463-477 (2019).

Jansen ME et al, “Effect of contact lens wear and a near task on tear film break-up,” *Optom Vis Sci.* 87:350-7 (2010).

Jiang X et al, “Capturing and evaluating blinks from video-based eyetrackers,” *Behav. Res. Methods* 45, 656–663 (2013).

Kato H et al, “Relationship between ocular surface epithelial damage, tear abnormalities, and blink in patients with dry eye,” *Cornea* 38:318–324 (2019).

Korb DR et al, “Tear film lipid layer thickness as a function of blinking,” *Cornea* 13:354-9 (1994).

Lemp MA & Hamill JR, “Factors affecting tear film breakup in normal eyes,” *Arch Ophthalmol* 89:103-5 (1973).

Li M et al, “Assessment of vision-related quality of life in dry eye patients,” *Invest Ophthalmol Vis Sci* 53(9):5722-5727 (2012).

Mitelman R et al, “A noninvasive, fast and inexpensive tool for the detection of eye open/closed state in primates,” J. Neurosci. Methods 178, 350–356 (2009).

Nakamori K et al, “Blinking is controlled primarily by ocular surface conditions,” Am J Ophthalmol. 124:24-30 (1997).

Norn M, “Desiccation of the precorneal tear film I. Corneal wetting time,”. Acta Ophthalmol 47:865-80 (1969).

O’Gorman L, Sammon MJ, Seul M, ‘Practical Algorithms for Image Analysis’, 2nd Edition, Cambridge University Press, Cambridge UK (2008).

Patel S. et al, “Effect of visual display unit use on blink rate and tear stability” Optom Vis Sci. 68:888-92 (1991).

Petrou M & Petrou C, ‘Image Processing the Fundamentals’, Wiley-Blackwell, London UK, 2nd Edition, (2010).

Pult H et al, “The relation between blinking and conjunctival folds and dry eye symptoms,” Optom Vis Sci. 90(10):1034-9 (2013).

Rosenfield M, “Computer vision syndrome: a review of ocular causes and potential treatments,” Ophthalmic Physiol Opt. 31(5):502-515 (2011).

Rosenfield M, “Vision and IT displays: a whole new visual World” Ophthalmic Physiol Opt 32:363-6 (2012).

RussJC. The image processing handbook CRC press (2016).

Sweeney DFM et al, "Tear film stability: a review," *Exp eye Res* 2013;117:28-38 (2013).

Schlote T et al, "Marked reduction and distinct patterns of eye blinking in patients with moderately dry eyes during video display terminal use," *Graefes Arch Clin Exp Ophthalmol*. 242:306-12 (2004).

Tan H & Zhang YJ, "Detecting eye blink states by tracking iris and eyelids," *Pattern Recognit. Lett.* 27, 667–675 (2006).

Tian Yet al, "Dual-state parametric eye tracking," in *Proc. Fourth IEEE Int. Conf. Automatic Face and Gesture Recognition*, pp. 110–115 (2000).

Tosi S. 'Matplotlib for Python developers', Packt publishing Ltd. (2009).

Tsubota K & Nakamori K, "Dry eyes and video display terminals," *New Engl J Med* 328:584 (1993).

Uchino M et al, "Prevalence of dry eye disease among Japanese visual display terminal users. *Ophthalmology*," 115:1982–1988 (2008).

Vitali C et al, "Classification criteria for Sjogren's syndrome: a revised version of the European criteria proposed by the American-European Consensus Group," *Ann Rheum Dis* 61:554-8 (2002).

Wang TMT et al, "Impact of blinking on ocular surface and tear film," *Ocul Surf*. 16(4):424-429 (2018).

Widodo CE et al. 'Medical image processing using python and open cv' *J. Phys.: Conf. Ser.* 1524; 012003 (2020).

Won OL et al, “Blink detection robust to various facial poses,” *J. Neurosci. Methods* 193, 356–372 (2010).

Yap M, “Tear break-up time is related to blink frequency,” *Acta Ophthalmol (Copenh)*. 69:92-4 (1991).

Yilmaz U et al, ‘Dry eye disease and depression-anxiety-stress: A hospital-based case control study in Turkey’, *Pak J Med Sci*. 31(3):626-631 (2015).

Young IT, Gerbrands JJ, Viliet LJV. ‘Fundamentals of Image Processing’, Delft University of Technology (2007).

EK 1. GÖZ KIRPMA SAYISINI ÖLÇEN PYTHON KODU

```
import cv2
from cv2 import CascadeClassifier

yuz_cascade: CascadeClassifier =
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
goz_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

kamera = cv2.VideoCapture(0)
c=0

while True:
    ret,img = kamera.read()
    gri_ton = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    yuzler = yuz_cascade.detectMultiScale(gri_ton, 1.3, 5)
    for (x, y, w, h) in yuzler:
        k=1
        cv2.rectangle(img, (x,y), (x+w, y+h), (255,0,0), 2)
        roi_gray = gri_ton[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        gozler = goz_cascade.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in gozler:
            k=0
            cv2.rectangle(roi_color, (ex,ey), (ex+ew,ey+eh), (0,255,0), 2)
        if k==1:
            out=c,"DEFA GOZ KIRPTINIZ"
            c=c+1
            print(out)
        else:
            print("GOZ KIRPMIYORSUNUZ")
    cv2.imshow('Goz_Kirpma_Sayaci', img)
    k = cv2.waitKey(30) & 0xff
    if k==27:
        break
kamera.release()
cv2.destroyAllWindows()
```

