

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

GÖRÜNTÜ İŞLEME VE BLOKZİNCİRİ TABANLI BİR
İYİLEŞTİRİLMİŞ OPTİK OKUMA VE SINAV
DEĞERLENDİRME SİSTEMİ

YÜKSEK LİSANS TEZİ

SERCAN SARP

DANIŞMAN

DOÇ. DR. SELÇUK TOPAL

AĞUSTOS 2024

BİTLİS

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
MATEMATİK ANABİLİM DALI

GÖRÜNTÜ İŞLEME VE BLOKZİNCİRİ TABANLI BİR
İYİLEŞTİRİLMİŞ OPTİK OKUMA VE SINAV
DEĞERLENDİRME SİSTEMİ

YÜKSEK LİSANS TEZİ

SERCAN SARP

ORCID 0000-0002-1621-9651

DANIŞMAN

DOÇ. DR. SELÇUK TOPAL

AĞUSTOS 2024

BİTLİS

T.C.

BİTLİS EREN ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZ ÇALIŞMASI

ETİK BEYANI

Lisansüstü Eğitim Enstitüsü Matematik Anabilim Dalı Yüksek Lisans öğrencisiyim. Hazırlamış olduğum “**Görüntü İşleme ve Blokzinciri Tabanlı Bir İyileştirilmiş Optik Okuma ve Sınav Değerlendirme Sistemi**” başlıklı tezde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi; tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu; tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi; kullanılan verilerde herhangi bir değişiklik yapmadığımı; bu tezde sunduğum çalışmanın özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim. .../08/2024

İmza

Sercan SARP

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
TEZ YAZIM KILAVUZU UYGUNLUK BEYANI

“Görüntü İşleme ve Blokzinciri Tabanlı Bir İyileştirilmiş Optik Okuma ve Sınav Değerlendirme Sistemi” başlıklı yüksek lisans tezi Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kılavuzuna uygun olarak hazırlanmıştır.
.../08/2024

Tezi Hazırlayan

Danışman

İmza

İmza

Sercan SARP

Doç. Dr. Selçuk TOPAL

Matematik Anabilim Dalı Başkanı

İmza

Doç. Dr. Derya ARSLAN

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
TEZ ONAY SAYFASI

Sercan SARP tarafından hazırlanan “**Görüntü İşleme ve Blokzinciri Tabanlı Bir İyileştirilmiş Optik Okuma ve Sınav Değerlendirme Sistemi**” adlı tez çalışması, .../08/2024 tarihinde yapılan sınavla aşağıdaki jüri tarafından oy birliği/ oy çokluğu ile Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Matematik Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul/ red edilmiştir.

JÜRİ:

İMZA

Danışman: Doç. Dr. Selçuk TOPAL
Gebze Teknik Üniversitesi

Üye: Doç. Dr. Derya ARSLAN
Bitlis Eren Üniversitesi

Üye: Dr. Öğr. Üyesi Ferhat TAŞ
İstanbul Üniversitesi

Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Yönetim Kurulunun.../.../.... tarih ve sayılı kararıyla jüri tarafından kabul edilmiş bu çalışmanın yüksek lisans tezi olarak kabulü onaylanmıştır.

...../...../.....

Doç. Dr. Mehmet Bakır ŞENGÜL
Enstitü Müdürü

T.C.

Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü

Matematik Anabilim Dalı

**GÖRÜNTÜ İŞLEME VE BLOKZİNCİRİ TABANLI BİR İYİLEŞTİRİLMİŞ
OPTİK OKUMA VE SINAV DEĞERLENDİRME SİSTEMİ**

Yüksek Lisans Tezi

Sercan SARP

Danışman: Doç. Dr. Selçuk TOPAL

Ağustos 2024

ÖZET

Bu tezde, sınav sistemlerinin güvenliğini artırmak ve adayların sınav sürecinin bütünlüğüne olan güvenini güçlendirmek için görüntü işleme ve blok zinciri teknolojisini entegre eden bir model önerilmiştir. Önerilen sistem, sınav öncesi optik okuma ve değerlendirmeyi içerir ve adayların sınavdan önce doğru ve yanlış cevap sayısını gözden geçirmelerine olanak tanır. Sistem iş akışı, adayların sınav binasına girmeden önce kameralar tarafından fotoğraflarının çekilmesiyle başlar. Bu görüntüler, doğru tanımlamayı sağlamak için bir bilgisayarda çalışan bir algoritma tarafından işlenir. Benzer şekilde, sınavdan sonra adaylar tamamlanmış optik formlarıyla bir kameranın önüne çıkar ve daha sonra bunlar önceden belirlenmiş bir formatta yakalanır. Sistem, arka uç algoritmaları kullanarak optik okuma ve yüz tanıma gerçekleştirir ve tüm ilgili veriler blokzincirinde güvenli bir şekilde kaydedilir ve bir veritabanında saklanır. Bu mekanizma, aday yanıtlarının yetkisiz kişiler tarafından değiştirilmesini önler ve böylece sınav sürecinin güvenliğini ve güvenilirliğini artırır. Bu çalışma, sınav yönetiminin bütünlüğünü ve güvenilirliğini güçlendirmeyi amaçlayan yeni bir çözüm sunmaktadır.

Anahtar kelimeler: Algoritma, Blokzincir, Görüntü İşleme, Makine Öğrenimi, Mantık

Republic of Türkiye

Bitlis Eren University Graduate School

Department of Mathematics

**AN IMPROVED OPTICAL READING AND EXAM EVALUATION SYSTEM
BASED ON IMAGE PROCESSING AND BLOCKCHAIN**

Master Thesis

Sercan SARP

Supervisor: Assoc. Prof. Dr. Selçuk TOPAL

July 2024

ABSTRACT

In this thesis, a model integrating image processing and blockchain technology is proposed to enhance the security of examination systems and strengthen candidates' confidence in the integrity of the exam process. The proposed system includes optical scanning and assessment before the exam, allowing candidates to review the number of correct and incorrect answers before the exam. The workflow of the system begins with candidates having their photos taken by cameras before entering the exam building. These images are processed by an algorithm running on a computer to ensure correct identification. Similarly, after the exam, candidates stand in front of a camera with their completed optical forms, which are then captured in a predetermined format. The system performs optical scanning and facial recognition using backend algorithms, and all relevant data is securely recorded on the blockchain and stored in a database. This mechanism prevents unauthorized alteration of candidate responses, thereby increasing the security and reliability of the exam process. This study presents a novel solution aimed at enhancing the integrity and reliability of exam management.

Keywords: Algorithm, Blockchain, Image Processing, Machine Learning, Logic

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde bana kıymetli bilgilerini paylaşan, her zaman sabırla ve ilgiyle yardımcı olan, sorun yaşadığımda yanına gidebildiğim, güler yüzünü ve samimiyetini esirgemeyen kıymetli danışman hocam **Doç. Dr. Selçuk TOPAL**’ a içten teşekkürlerimi sunuyorum. Gelecekteki meslek hayatımda da verdiği bilgilerin ışığında yürüyeceğimi biliyorum ve bu desteği için minnettarım.

Bir öğretmen olarak, öğrenim hayatım boyunca karşılaştığım tüm öğretmenlerime bıraktıkları izler için teşekkür ediyorum. Beni gelecekte söz sahibi yapacak bilgilerle donattıkları için hepsine saygılarımı iletiyorum.

Özellikle bu süreçte her zaman desteklerini yanımda hissettiğim değerli eşim **İrem SARP**’ a ve bana sevgi ve saygı kavramlarını öğreten, hiçbir zaman desteklerini esirgemeyen başta annem **Sevim SARP**, babam **Mustafa SARP** ve kardeşlerim olmak üzere tüm aileme sonsuz teşekkürlerimi iletiyorum.

İÇİNDEKİLER

ÖZET.....	I
ABSTRACT	II
TEŞEKKÜR	III
İÇİNDEKİLER	IV
ŞEKİLLER DİZİNİ	V
ÇİZELGELER DİZİNİ	VII
FOTOĞRAFLAR DİZİNİ	VIII
KISALTMALAR.....	IX
GİRİŞ	1

BİRİNCİ BÖLÜM

GÖRÜNTÜ İŞLEME

1.1. Görüntü İşleme Alanında Yapılan Çalışmaların Analizi	5
1.2. Görüntü İşleme Modülleri	6
1.2.1. Yüz Tanıma Modülü.....	8
1.2.2. Optik Okuma ve Değerlendirme Modülü.....	15

İKİNCİ BÖLÜM

BLOKZİNCİR

2.1. Blokzincir Alanında Yapılan Çalışmaların Analizi	22
2.2. Blokzincir Modülleri	24
2.2.1. Kullanıcı Dostu Arayüzün Tasarlanması	25
2.2.2. Görüntü Verilerinin Blokzincirde Depolanması.....	31
2.2.3. Verilerin Şifrelenmesi.....	35

ÜÇÜNCÜ BÖLÜM

AKILLI SINAV GÜVENLİĞİ

3.1. Akıllı Sınav Güvenliği Alanında Yapılan Çalışmaların Analizi	42
3.2. Akıllı Sınav Güvenliği Sisteminin Uygulama Aşamaları.....	43
SONUÇ.....	51
KAYNAKLAR.....	52

ŞEKİLLER DİZİNİ

Şekil 1.2.1.1. Yüz tanıma için veri tabanı oluşturulması.....	10
Şekil 1.2.1.2. Yüz tanıma için veri tabanı oluşturulması-2.....	10
Şekil 1.2.1.3. Yüz tanıma için veri tabanı oluşturulması-3.....	11
Şekil 1.2.1.4. Yüz tanıma için veri tabanı oluşturulması-4.....	11
Şekil 1.2.1.5. Yüz tanıma için veri tabanı oluşturulması-5.....	12
Şekil 1.2.1.6. Yüz tanıma için veri tabanı oluşturulması-6.....	13
Şekil 1.2.1.7. Yüz tanımanın gerçekleştirilmesi	13
Şekil 1.2.1.8. Yüz tanımanın gerçekleştirilmesi-2.....	14
Şekil 1.2.1.9. Yüz tanımanın gerçekleştirilmesi-3.....	14
Şekil 1.2.1.10. Yüz tanımanın gerçekleştirilmesi-4.....	14
Şekil 1.2.2.1. Optik okumanın gerçekleştirilmesi.....	16
Şekil 1.2.2.2. Optik okumanın gerçekleştirilmesi-2	17
Şekil 1.2.2.3. Optik okumanın gerçekleştirilmesi-3	17
Şekil 1.2.2.4. Optik okumanın gerçekleştirilmesi-4	18
Şekil 1.2.2.5. Optik okumanın gerçekleştirilmesi-5	18
Şekil 1.2.2.6. Optik okumanın gerçekleştirilmesi-6	18
Şekil 1.2.2.7. Optik okumanın gerçekleştirilmesi-7	19
Şekil 1.2.2.8. Optik okumanın gerçekleştirilmesi-8	19
Şekil 1.2.2.9. Optik okumanın gerçekleştirilmesi-9	20
Şekil 1.2.2.10. Optik okumanın gerçekleştirilmesi-10	20
Şekil 1.2.2.11. Optik okumanın gerçekleştirilmesi-11	21
Şekil 1.2.2.12. Optik okumanın gerçekleştirilmesi-12	21
Şekil 1.2.2.13. Optik okumanın gerçekleştirilmesi-13	21
Şekil 2.2.1.1. Kullanıcı dostu arayüzün tasarlanması	27

Şekil 2.2.1.2. Kullanıcı dostu arayüzün tasarlanması-2.....	27
Şekil 2.2.1.3. Kullanıcı dostu arayüzün tasarlanması-3.....	27
Şekil 2.2.1.4. Kullanıcı dostu arayüzün tasarlanması-4.....	28
Şekil 2.2.3.1. Görüntü verilerinin şifrelenmesi.....	37
Şekil 2.2.3.2. Görüntü verilerinin şifrelenmesi-2	37
Şekil 2.2.3.3. Görüntü verilerinin şifrelenmesi-3	37
Şekil 2.2.3.4. Görüntü verilerinin şifrelenmesi-4	38
Şekil 2.2.3.5. Görüntü verilerinin şifrelenmesi-5	38
Şekil 2.2.3.6. Görüntü verilerinin şifrelenmesi-6	39
Şekil 2.2.3.7. Görüntü verilerinin şifrelenmesi-7	39
Şekil 3.2.1. Sınav anı uygulamaları akış diyagramı.....	44
Şekil 3.2.2. Sınav sonrası uygulamaları akış diyagramı	45

ÇİZELGELER DİZİNİ

Çizelge 1.1. Blokzincirin kullanım alanları 1

Çizelge 1.2. Görüntü işlemenin kullanım alanları 3

FOTOĞRAFLAR DİZİNİ

Fotoğraf 1.2.1. Scikit-image ile yapılmış bir görüntü işleme	7
Fotoğraf 1.2.1.1. Derin Öğrenme Ağırlık Modelindeki 68 nokta	9
Fotoğraf 1.2.2.1. OpenCV kütüphanesinin dokümantasyonundan bir örnek.....	15
Fotoğraf 2.1.1. Blokzincir kullanım alanları	22
Fotoğraf 2.2.1.1. Arayüz giriş sayfası	28
Fotoğraf 2.2.1.2. Ana sayfa ekranı.....	29
Fotoğraf 2.2.1.3. Aday bilgileri ekranı.....	29
Fotoğraf 2.2.1.4. Sonuç açıklama ekranı	30
Fotoğraf 2.2.1.5. Sistem geri bildirimi ve destek ekranı.....	30
Fotoğraf 2.2.2.1. Uygulamanın giriş ekranı	32
Fotoğraf 2.2.2.2. Ganache test hesabı ile Metamask bağlantısı.....	33
Fotoğraf 2.2.2.3. Dosya yükleme ve onay ekranı	33
Fotoğraf 2.2.2.4. Yüklenen dosya bilgilerinin görüntülenmesi	34
Fotoğraf 2.2.2.5. Ganache hesabı üzerinden yapılan işlemin görüntülenmesi	35
Fotoğraf 2.2.3.1. Görüntü verisini şifreleme işleminin görüntülenmesi	40
Fotoğraf 2.2.3.2. Görüntü verisinin şifre çözme işleminin görüntülenmesi	40
Fotoğraf 2.2.3.3. Başarısız olan bir şifre çözme işlemi	41
Fotoğraf 2.2.3.4. Ganache hesabı üzerinden şifreleme işleminin görüntülenmesi	41
Fotoğraf 3.2.1. 68 noktalı yüz tanıma modeli ve maskeli yüz tanıma	46
Fotoğraf 3.2.2. Yüz tanıma modeli ile ikizlerin ayırt edilmesi	46
Fotoğraf 3.2.3. Standart optik formun görüntüsü.....	47
Fotoğraf 3.2.4. Standart optik form üzerinde yapılan görüntü işleme aşamaları.....	48
Fotoğraf 3.2.5. Aday yüz tanınması ve optik formunun değerlendirilmesi	48
Fotoğraf 3.2.6. Giriş yapan herhangi bir kişinin görüntüleyeceği ekran	49
Fotoğraf 3.2.7. Sınav yetkililerinin görüntüleyeceği ekran	50
Fotoğraf 3.2.8. Sınava giren adayın görüntüleyeceği ekran	50

KISALTMALAR

AES	: Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
Bkz	: Bakınız
CSS	: Cascading Style Sheet (Basamaklı Stil Sayfaları)
De-Fi	: Decentralized Finance (Merkeziyetsiz Finans)
DES	: Data Encryption Standard
DISP	: Distributed Image Storage Protocol (Dağıtık Görüntü Depolama Protokolü)
ECDSA	: Elliptic Curve Digital Signature Algorithm (Eliptik Eğri Dijital İmza)
FCNN	: Fuzzy Convolutional Neural Network (Bulanık Evrişimsel Sinir Ağları)
HOG	: Histogram of Oriented Gradients (Yönlendirilmiş Gradyanların Histogramı)
HTML	: HyperText Markup Language (Hiper Metin İşaretleme Dili)
IoT	: Internet of Things (Nesnelerin İnterneti)
IoMT	: Internet of Medical Things (Medikal Nesnelerin İnterneti)
IPFS	: InterPlanetary File System (Gezegenler Arası Dosya Sistemi)
LBPH	: Local Binary Patterns Histograms (Yerel İkili Modeller Histogramları)
NFT	: Nitelikli Fikri Tapu
P2P	: Peer-to-peer (Eşler Arası)
PoW	: Proof of Work (İş Kanıtı)
RSA	: Rivest-Shamir-Adleman

GİRİŞ

Yakın gelecekte interneti kullanan cihaz sayısında gerçekleşecek muhtemel artışla birlikte, bu cihazların hemen hepsinde blokzincir teknolojisi kullanılabilir olacaktır. Bu da blokzincirin kullanım alanını günden güne yaygınlaştıracaktır. Bu artış beraberinde yeni ihtiyaçları meydana getirecektir. İşte burada nesnelerin interneti (IoT) denilen kavramı yani interneti kullanan her türlü aracı bu alana dahil edebiliriz. Akıllı telefonlar, akıllı ev aletleri, akıllı tarım sistemleri, akıllı prizler, akıllı kilitler gibi bir şekilde internete erişerek kişilerle veya diğer cihazlarla iletişim halinde olan bu sistemler blokzincirin temel özelliklerinden faydalanarak, işlevlerini artıracaklardır [1].

Çizelge 1.1. Blokzincirin kullanım alanları

Kullanım Alanları	Blokzincir kullanımının faydaları
Tedarik Zinciri	Blokzincir, talebi en kısa sürede ve düşük maliyetle karşılamayı, şirketlerin ürünlerinin tedarik sürecini dijital bir veri tabanında takip etmelerini ve düzenli olarak aksiyon almalarını sağlıyor [2].
Akıllı Sözleşmeler	Çoğunlukla dijital para birimlerinin(token) altyapısında kullanılan blokzincir tabanlı akıllı sözleşmeler, her alanda merkezi olmayan uygulamaların geliştirilmesine olanak sağlıyor.
Telif Hakları	Blokzincir temelde bir veri tabanı olduğundan sanat eserleri ve (NFT) Nitelikli Fikrî Tapuların çeşitli verilerini (telif hakları, sanatçının kimliği, eserin tarihi vb.) içerir.
Kişisel Kimlik	Blokzincir tıpkı bir sosyal kimlik gibi insanlara ait tüm bilgileri saklayabilir ve bilgiler değiştirilemez (isim, fotoğraf, deneyim, sertifika bilgileri vb.).
Finans Sektörü	Merkeziyetsiz Finans (De-Fi), merkezi yapıyı ortadan kaldırıp karar alma işini dağıttığı için oldukça güvenilirdir. Blokzincir bu dağıtık yapının oluşturulmasında oldukça önemlidir.
Büyük Verilerin Depolanması	Blokzincir, merkezi bir sunucu ihtiyacı olmayan P2P (Peer-To-Peer) yani eşler arası ağdaki bilgisayarlar tarafından yönetilir. Bu yönetim şekli de daha yüksek güvenlik, şeffaflık ve esneklik sağlayarak büyük verilerin depolanmasını ve işlenmesini kolaylaştırır [3].

Blokzincirin yaygın kullanım alanı kripto paralar gibi görünse de **Çizelge 1. 1**'de blokzincir yapısının diğer kullanım alanları görülmektedir. Tabi ki kullanım alanları **Çizelge 1.1.** ' de yazılanlarla da sınırlı değildir. Yukarıda yazılan kullanım alanları genişledikçe avantajları artacak, avantajları arttıkça kullanım alanları genişleyecektir. Gizlilik, güvenlik, düşük transfer ücretleri, hızlı işlemler, uzun ömürlülük ve güvenilirlik, izlenebilirlik, değiştirilemezlik ve şeffaflık gibi özellikleri birleşerek yeni modeller oluşabilir.

Zararlarına veya dezavantajlı olduğu noktalara gelecek olursak; kötü niyetli saldırılar, verilerin değiştirilemez olması, özel anahtarlar, verimsizlik ve büyük verileri depolama sıkıntısı konularını dezavantaj sayabiliriz. Özellikle PoW (Proof-of-Work) algoritmasının kullanıldığı durumlarda ağdaki hash işlem gücünün yüzde 51'ini elinde bulunduran bir birim kötü niyetli saldırılar yapabilir. Bu kâğıt üstünde mümkün görünse de pratikte ağ büyüdükçe güvenlik de artacağından çok düşük ihtimalden öteye gidemeyecektir [4].

Her ne kadar değiştirilemezlik blokzincirin avantajları içinde yazılmış olsa da bazı durumlarda değiştirilemezlik dezavantaja dönüşebilir. Çünkü veri bir kere blokzincire eklendiği zaman değiştirmek için genelde zincirde bulunan geçersiz blokları ve işlemleri geçerli hale getirerek yeni bir sistem oluşturan hard fork'u gerektirir. Kullanıcının cüzdanına erişebilmesi için 2 adet anahtarı vardır. Açık anahtar ve özel anahtar. Açık anahtarı herkesle paylaşılabilse de özel anahtar kaybedilince kullanıcı hesabına erişimi yani parasını kaybetmiş olur [5].

Verimsizlik noktasında ise, özellikle PoW algoritmasını kullananlar verimsizdir. Algoritma gereği madenciler işlem gücü kadar ödül kazanacaklar ve işlem güçleri ne kadar yüksek olursa o kadar fazla enerji tüketecekler [6].

En önemli avantajlarından olduğu gibi dezavantajlardan bir tanesi ise depolama olacaktır. Blokzincir kayıt defterleri zaman ile devasa boyutlara ulaşarak depolanması zorlaşacaktır. Depolama zorlaştıkça ise ağ nodelarını (düğüm) kaybedecek ve sistem sıkıntıya girecektir [3].

Blokzincir, kullanım alanları, avantajları, dezavantajlarından kısaca bahsedildi. Sırada sistemin diğer modülü olan görüntü işleme var. Görüntü işlemeyi temel olarak cisimlerin kamera veya sensörler tarafından algılanıp; algoritma ile çalışan bilgisayarda işlenmesiyle birlikte görüntü çıktısı alınması olarak tanımlayabiliriz.

Çizelge 1.2. Görüntü işlemenin kullanım alanları

Kullanım Alanları	Görüntü İşlemenin Kullanım Alanları [7]
Sağlık Endüstrisi	Tıbbi görüntüleme teknikleri ile elde edilen görüntülerin işlenmesiyle vücuttaki mevcut kusurların onarılması veya iyileştirilmesi sağlanmaktadır [8].
Trafik	Özellikle nüfusun yoğun olduğu bölgelerde trafik akışının kontrol edilmesi ve plaka tanınması için sıklıkla görüntü işleme teknikleri kullanılmaktadır [9].
Tarım	Tarım makinelerinin işleyişinin kontrol edilip, olası aşınmaların tespiti ve yetiştirilen ürünlerin kalitesini uzaktan takip edebilmek amacıyla kullanılmaktadır [10].
Tasarım ve İmalat	Orjinal ve güvenilir ürünlerin tasarımı için bozulmuş görüntüleri işlemek ve radyometrik düzeltmeleri gerçekleştirmek, görüntü işlemenin öne çıkan kullanım alanlarından [11].
Savunma Endüstrisi	Uzaktan algılanan araziler, otonom askeri araçlar, silah dürbünleri ve güdümlü roketler gibi çeşitli uygulama alanlarında kullanılmaktadır [12].
Güvenlik	Özellikle veri ve görüntü şifreleme (Steganografi) alanında, kamerada görülen nesnenin tespiti, tanınması ve doğrulanması işlemlerinde kullanılmaktadır [13].
Yüz Tanıma	Farklı makine öğrenmesi teknikleri ve ilgili yazılım dillerinin kütüphaneleri kullanılarak geliştirilen yüz tanıma sistemleri, siber ve fiziksel güvenlik açısından büyük önem taşımaktadır [14].
Optik Okuma ve Değerlendirme	Optik işaret tanıma sistemleri özellikle fiziksel testler ve sınavlarda kullanılmaktadır. Bunun dışında işaret tanıma olarak değerlendirildiğinde daha geniş kullanım alanları oluşacaktır [15].

Çizelge 1.1. ' de blokzincirin, Çizelge 1.2. ' de ise görüntü işlemenin kullanım alanlarından ve özelliklerinden bahsedilmiştir. Tüm bu kullanım alanları elbette ki bu iki terimi de tanımlamak için yeterli olmayacaktır.

Bu alanlar çoklu olarak birleştirildiğinde ortaya çok daha efektif çalışmalar çıkacaktır. Örneğin trafik uygulama alanında kullanılan plaka tanıma sistemi ile yüz tanıma sistemi bir arada kullanılıp hem araç hem de sürücü takip edilerek yasal işlemler yapılabilir. Ek olarak blokzincir teknolojisi ile entegrasyonu da yapıldığında görüntülerin dağıtık mimari ile değiştirilmeden saklanması sağlanabilir. Başka bir yaklaşım olarak Blokzincir ile tasarlanan bir tedarik zinciri yardımıyla, tarım alanında görüntü işleme yöntemleriyle tespit edilen aksaklıklar giderilebilir.

Görüldüğü üzere Blokzincir ve Görüntü İşleme teknolojisi birlikte kullanıldığında çok daha verimli modeller oluşturulabilmektedir. Blokzincir ve Görüntü İşleme Tabanlı olarak geliştirilecek bir sınav sisteminin en büyük avantajı güvenlik ve gizlilik olacaktır. Yapılan literatür taraması ile bu alanların her birinde az sayıda çalışma bulunması getirilen çözüm önerisini ön plana çıkarmaktadır.

- **Görüntü İşleme:** görüntüler üzerinde birtakım işlemler yaparak istenen sonuçları elde etmeye olanak sağlayan teknolojidir. Dijital veri analizi alanlarında sıklıkla kullanılır.
- **Blokzincir:** günümüzün en önemli teknolojilerinden biri olup, kullanım şekillerine göre farklılıklar arz edebilir. Gizlilik ve güvenlik yaygın kullanım amacıdır.
- **Sınav Güvenliği:** çeşitli yöntemler aracılığıyla gerek sınavın gerekse adayın güvenliğini sağlamak amacıyla geliştirilen her türlü sistemdir. Yukarıdaki tanımlamalara baktığımızda blokzincir ve görüntü işlemeyi kullanmanın doğal sonucu olarak sınav güvenliğinin artacağını öngörülmektedir.

Bu çalışmada birinci bölümde Görüntü İşleme konusu detaylıca işlenecektir. İkinci bölümde Blokzincir ve uygulama örnekleri paylaşılacaktır. Üçüncü bölümde ise akıllı sınav güvenliğinin blokzincir ve görüntü işleme ekseninde nasıl sağlanacağını üzerinde durulacaktır.

BİRİNCİ BÖLÜM

GÖRÜNTÜ İŞLEME

1.1. Görüntü İşleme Alanında Yapılan Çalışmaların Analizi

Bu bölümde Görüntü İşleme ve Analizi alanında geçtiğimiz yıllarda yapılan çalışmalar, bunların hitap ettiği alanlar ve literatüre yaptıkları katkılardan kısaca bahsedilecektir.

2021 yılında yapılan Pikselden Görüntü Analizine konulu çalışmada radyolojik görüntüleme alanında, uzmanların işini kolaylaştırmak amacıyla bilgisayar destekli makine öğrenimi kullanılmaktadır [16].

Yine 2020 yılında yayınlanan bir çalışmada Görüntü Algılama ve Blokzincir Depolama konusunda işlenen görüntü bilgilerinin, blokzincir teknolojisi aracılığıyla güvenli bir şekilde aktarılıp ve saklanabileceğinin üzerinde durulmuştur [17].

Aynı yıllarda yürütülen bir çalışmada Görüntüler için Blockchain Teknolojisinin Uygulanması konusu irdelenmiştir. Bu çalışmada dijital görüntüleri hash bağlantılarıyla zincirleyen ve blokzincirden farklı olarak görüntülerin blokların içinde saklanmadığı bir kriptografik yapı önerisi sunulmuştur [18].

Çoklu Blokzincire Dayalı Tıbbi Görüntü Analizi için Gizliliği Koruyan İşbirlikçi Eğitim Modeli isimli çalışmada ise, veri gizliliği, özellikle hasta tıbbi verilerinin gizliliğinin sağlanması için, araştırmacıların özel verilerini ifşa etmeden işbirlikçi eğitim yapmalarını etkili bir şekilde sağlayacak bir yaklaşım geliştirilmiştir [19].

Görüntü depolama için dağıtılmış havuzlama blokzincir protokolüne sahip bir model üzerine yazılan bir makalede, Gezegenlerarası Dosya Paylaşım Sistemi (IPFS) kullanarak blokzincir depolama alanını etkili bir şekilde artıran ve hesaplama maliyetlerini düşüren Dağıtılmış Görüntü Depolama Protokolü'ne (DISP) dayalı bir protokol önerilmektedir; görüntü kanıtlarını küçük dosyalara bölerek ve dağıtılmış düğümlerde depolayarak büyük ölçekli endüstriyel ve ticari uygulamalar için uygun koşullar yaratmaktadır [20].

Bulanık Konvolüsyonel Sinir Ağı Kullanılarak Bulut ve Nesnelerin İnterneti Tabanlı Tıbbi Görüntü Teşhisinin Güvenliği ve Gizliliği konulu çalışma, genişletilmiş zigzag görüntü şifreleme şeması ve bulanık konvolüsyonel sinir ağı (FCNN) algoritması

kullanarak güvenli bir bulut bilişim ortamında hastalık sınıflandırması yapmayı amaçlamaktadır [21].

Yapılan literatür taramaları sonucunda görüntü işleme alanında henüz sayıca fazla araştırma yapılmamış olduğu göze çarpmaktadır. Bu alanda yapılacak yeni çalışmalar literatüre katkı sağlayacaktır. Halihazırda yapılmış çalışmaları çoğu ise sağlık alanındadır. Sağlık biliminde özellikle radyolojik görüntüleme tekniklerinin çok olması nedeniyle işlenebilecek görüntü sayısı da o oranda fazladır.

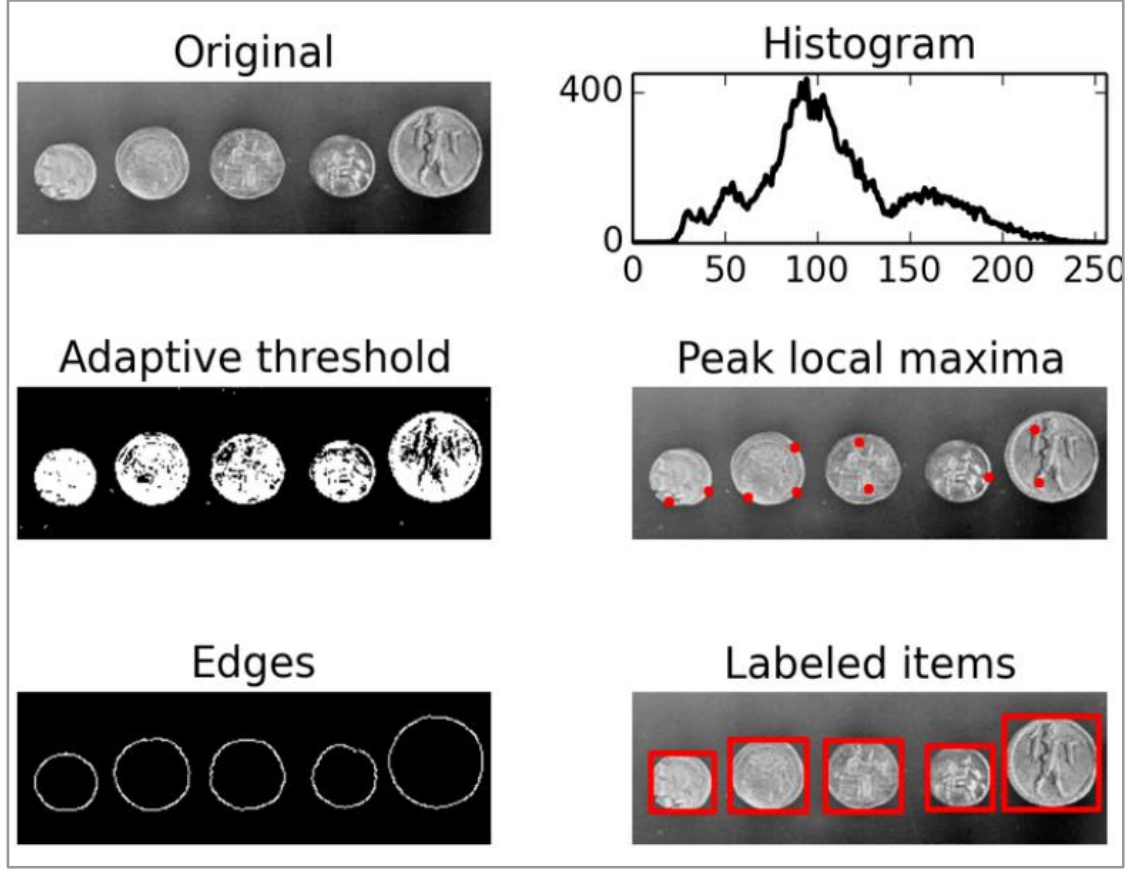
Görüntü işleme, yalnızca temel görüntü işleme tekniklerinden ibaret değildir. Aynı zamanda, görüntü tespiti, görüntü algılama, görüntü analizi, görüntü parçalama ve görüntü modelleme gibi birçok alt modülü de kapsamaktadır. Her bir alt modül, görüntü işleme sürecinin farklı yönlerine odaklanarak, elde edilen verilerin daha derinlemesine analiz edilmesini ve anlamlandırılmasını sağlar.

1.2. Görüntü İşleme Modülleri

Görüntü işleme, dijital görüntülerin işlenmesi ve analiz edilmesi sürecidir ve bilgisayarların insan görsel sistemini taklit ederek görüntüleri anlamlandırmasını ve yorumlamasını sağlar. Bu alandaki teknikler, görüntülerin iyileştirilmesi, segmentasyonu, özellik çıkarımı, tanınması ve sınıflandırılması gibi birçok uygulamayı kapsar. Bu uygulamalar temel olarak iki ana aşamadan oluşur, görüntülerin ön işlenmesi ve analiz edilmesi. Ön işleme aşamasında, görüntüler çeşitli filtreleme, iyileştirme ve gürültü giderme teknikleri kullanılarak işlenir. Bu aşama, görüntülerin kalitesini artırarak, analiz aşamasında daha doğru sonuçlar elde edilmesini sağlar. Analiz aşamasında ise, işlenmiş görüntülerden özellikler çıkarılarak nesne tanıma, sınıflandırma ve diğer analizler gerçekleştirilir.

Python programlama dilinin scikit-image ve NumPy kütüphanesi ile yapılmış olan bir görüntü işleme çalışması ve bu çalışmada bulunan bir görüntü **Fotoğraf 1.2.1.** olarak aşağıda paylaşılmıştır.

Bu kütüphanelerden scikit-image kütüphanesinde mevcut olan çeşitli fonksiyonlar, uyarlanabilir eşikleme (adaptive threshold), yerel maksimum algılama (peak local maxima), kenar tespiti ve etiketleme gibi işlemleri kapsamaktadır. Veri taşıyıcısı olarak NumPy dizileri kullanılarak yerleşik histogram fonksiyonlarının oluşturulmasına da olanak tanımaktadır.



Fotoğraf 1.2.1. Scikit-image ile yapılmış bir görüntü işleme [22].

Görüldüğü üzere uygulanan belli metotlar sayesinde basit kenar belirleme, işaretleme gibi işlemler yapılabilmektedir. Adaptive threshold fonksiyonu ile ön planı ve arka planı bölmek için ikili bir görüntü oluşturacak şekilde görüntü eşikleniyor. Çeşitli eşik algoritmaları mevcut olmakla birlikte burada, eşik değerinin bir pikselin yerel komşuluğu için ağırlıklı ortalama olduğu adaptive threshold fonksiyonu kullanılmıştır. Benzer şekilde yerel maksimumlar ve kenarlar gibi ilginç özellikler kolayca tespit edebilir. Peak Local Maxima fonksiyonu, bir görüntüdeki yerel maksimumların koordinatlarını belirlemek için kullanılabilir. Daha sonra, her bir parçaya ait olan bir alt resim çıkarmak için kullanılabilecek bir etiket atanmış olur. Son olarak, ölçüm fonksiyonu kullanılarak konum, alan, eksantriklik, çevre ve momentler gibi fiziksel bilgiler elde edilebilir [22].

Çalışmanın ilerleyen bölümlerinde OpenCV kütüphanesinin çalışma prensibi, ilgili görüntü işleme aşama ve modülleri tekrar inceleneceği için bu bölümde **Fotoğraf 1.2.1.**'deki örnek genel hatlarıyla anlamak için yeterli olacaktır.

1.2.1. Yüz Tanıma Modülü

Yüz tanıma, bireylerin yüz özelliklerini analiz ederek kimliklerini belirleme sürecidir. Teknolojinin ilerlemesiyle birlikte, yüz tanıma sistemleri güvenlik alanında olduğu kadar, eğitim ve sınav yönetimi gibi farklı sektörlerde de yaygın olarak kullanılmaktadır. Bu bölümde, yüz tanıma teknolojisinin sınav süreçlerindeki potansiyel uygulaması incelenecektir.

Yüz tanıma işlemi, görüntü işleme teknikleri kullanılarak gerçekleştirilecektir. İşlem sırasıyla şu aşamalardan oluşacaktır:

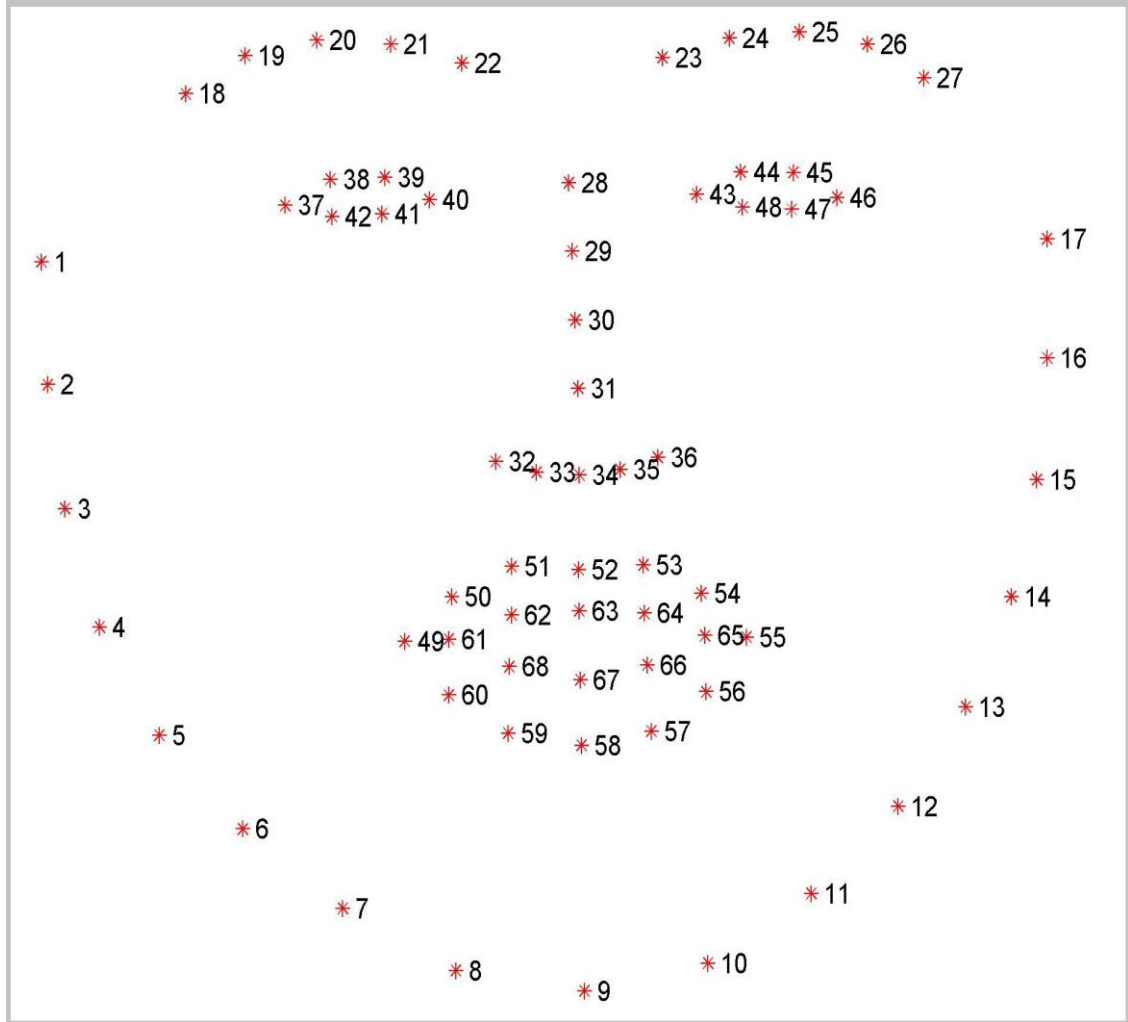
- Adayın yüzünün kamera ile fotoğraflanması
- Algoritma aracılığıyla fotoğraftaki yüzün hatlarını belirleme ve yüz özelliklerini çıkarma
- Önceden eğitilmiş bir yüz tanıma makine öğrenimi modeli kullanılarak adayın yüz özelliklerinin veri tabanındaki lerle karşılaştırılması ve eşleştirilmesi
- İşlemler sonucunda adayın kimliğinin doğrulanması veya reddedilmesi.

Bu işlem için birçok görüntü işleme kütüphanesi ve yüz tanıma algoritması mevcuttur. Özellikle, Python programlama dilinin açık kaynaklı kütüphanelerinden OpenCV (Açık kaynak kodlu bir görüntü işleme kütüphanesi) ve yüz tanıma için kullanılan algoritmalar arasında Eigenfaces, Fisherfaces ve Local Binary Patterns Histograms (LBPH) sıklıkla tercih edilmekle birlikte bu tezde shape-predictor- 68-face-landmarks derin öğrenme ağırlık modeline göre yorumlama yapılacaktır.

68 noktalı yüz tanıma modeli yüzün önemli bölgelerini (örneğin, gözler, burun, ağız, çene hattı gibi) temsil eden temel yüz hatlarını belirlemek için tasarlanmıştır. Bu noktalar, yüzün temel yapısını yüksek doğrulukla yakalayarak yüz tanıma ve ifadelerin tespit edilmesinde etkin bir şekilde kullanılabilir. Bu noktalardan elde edilen oranlar, yüzler arasındaki farkları ve benzerlikleri değerlendirmek için güçlü bir temel sunar.

Eğitilmiş model üzerinden yüz tanımayı yapmak için **Fotoğraf 1.2.1.1.**’ de görülen 68 noktayı kullanacağız. Bu 68 nokta belirlendikten sonra aralarındaki uzaklıklar ve oranlar algoritmada kaydedilerek akabinde tekrar eşleştirilerek değerlendirilecektir. Yüz tanıma veya tespit etme için OpenCV kütüphanesinin gerekli fonksiyonları kullanılabileceği gibi Dlib kütüphanesinin ilgili fonksiyonları da kullanılabilecektir. Kısacası Python programlama dilinin OpenCV ve Dlib kütüphaneleri makine öğrenimi,

derin öğrenme, bilgisayarla veri işleme ve görüntü işleme gibi çalışmalar yapmak için sıklıkla kullanılacaktır.



Fotoğraf 1.2.1.1. Derin Öğrenme Ağırlık Modelindeki 68 nokta

shape-predictor-68-face-landmarks modeline göre yorumlama yapılırken ilk olarak, bir veri girişi olan görüntü veya video içerisindeki yüzler algılanır. Yüz algılama işlemi, görüntüdeki yüzün yerini ve boyutunu belirlemek amacıyla genellikle HOG (Histogram of Oriented Gradients) tabanlı bir algoritma kullanır. Yüzü algıladıktan sonra, shape-predictor-68-face-landmarks modeli, yüzün içinde bulunan belirli anahtar noktaları (landmark) tespit eder. Bu landmarklar, yüzün gözleri, kaşları, burnu, ağzı ve diğer önemli yüz bölgelerini temsil eder.

Model, bu landmarkları koordinatları ile birlikte sunar. Model, büyük miktarda etiketlenmiş veri kullanılarak eğitilir. Bu eğitim süreci sırasında model, verileri işleyerek yüz landmark larını doğru bir şekilde tahmin etmeyi öğrenir. Daha sonra bu eğitilmiş model, yeni veri noktalarında yüz landmark larını tahmin etmek için kullanılır.

Öncelikle veri tabanı oluşturmak amacıyla adayların kayıt aşamasında çekilen fotoğraflarının yüklenmesi işlemiyle başlanır. Gerekli kütüphanelerin import edilmesi ilk işlem basamağı olacaktır. ‘face_recognition’ yüz tanıma işlemleri için, ‘cv2(OpenCV)’ görüntü işleme için ve ‘numpy’ ise sayısal işlemler için kullanılıyor. ‘cv2.VideoCapture(1)’ komutu ile ikinci kamera cihazından (1 numaralı cihaz) video yakalanıyor. Eğer harici kamera kullanılmıyorsa bu genellikle 0 numaralı cihaz olur.

```
import face_recognition
import cv2
import numpy as np
video_capture = cv2.VideoCapture(1)
```

Şekil 1.2.1.1. Yüz tanıma için veri tabanı oluşturulması

Bu kısımda, belirtilen dosya yolundaki görüntüler yükleniyor ve ‘face_recognition’ kütüphanesi kullanılarak her bir görüntüden yüz kodları çıkarılıyor. Yüz kodları, o yüzü tanımlayan ve karşılaştırmalarda kullanılan sayısal vektörlerdir.

```
sercan_image = face_recognition.load_image_file(r"C:\..\sercan.jpeg")
sercan_face_encoding = face_recognition.face_encodings(sercan_image)[0]

irem_image = face_recognition.load_image_file(r"C:\..\irem.jpeg")
irem_face_encoding = face_recognition.face_encodings(irem_image)[0]

selcuk_image = face_recognition.load_image_file(r"C:\..\selcuk.jpg")
selcuk_face_encoding = face_recognition.face_encodings(selcuk_image)[0]

arianna_image = face_recognition.load_image_file(r"C:\..\arianna.jpg")
arianna_face_encoding = face_recognition.face_encodings(arianna_image)[0]

brianna_image = face_recognition.load_image_file(r"C:\..\brianna.jpg")
brianna_face_encoding = face_recognition.face_encodings(brianna_image)[0]

adnan_image = face_recognition.load_image_file(r"C:\..\adnan.jpg")
adnan_face_encoding = face_recognition.face_encodings(adnan_image)[0]

afnan_image = face_recognition.load_image_file(r"C:\..\afnan.jpg")
afnan_face_encoding = face_recognition.face_encodings(afnan_image)[0]
```

Şekil 1.2.1.2. Yüz tanıma için veri tabanı oluşturulması-2

Bilinen yüz kodları ve bunlara karşılık gelen isimler iki ayrı listede tutuluyor. Bu listeler, yüz tanıma işleminde kullanılacak. Burada yüz kodlarına karşılık gelen değerler

isim, sınava girecek adayı temsil eden bir numara veya herhangi bir kod olabilir. Algoritma tarafından okunan yüz eğer tanınırsa, altında o kişiye karşılık verilen değer yazılacaktır.

```
known_face_encodings = [
    sercan_face_encoding,
    irem_face_encoding,
    selcuk_face_encoding,
    sarper_face_encoding,
    alper_face_encoding,
    arianna_face_encoding,
    brianna_face_encoding,
    adnan_face_encoding,
    afnan_face_encoding
]
known_face_names = [
    "Sercan S.",
    "irem SARP",
    "Selcuk T.",
    "Sarper S.",
    "Alper S.",
    "Arianna S.",
    "Brianna S.",
    "Adnan F.",
    "Afnan F."
]
```

Şekil 1.2.1.3. Yüz tanıma için veri tabanı oluşturulması-3

Yüz konumları, yüz kodları ve yüz isimleri gibi değişkenler başlatılıyor. ‘process_this_frame değişkeni’, her bir karede yüz işleme işlemlerinin yapılıp yapılmayacağını belirlemek için kullanılıyor.

```
face_locations = []
face_encodings = []
face_names = []
process_this_frame = True
```

Şekil 1.2.1.4. Yüz tanıma için veri tabanı oluşturulması-4

Sonraki aşamada sırasıyla şu işlemler yapılıyor.

- ‘while’ döngüsü, sürekli olarak video karelerini işlemek için kullanılıyor.

- Her karede, kameradan bir görüntü alınıyor.
- Performansı artırmak için her bir kare yerine her ikinci kare işleniyor.
- Görüntü boyutu küçültülerek işleme hızlandırılıyor (1/4 boyutuna küçültme).
- OpenCV BGR formatından 'face_recognition' tarafından kullanılan RGB formatına dönüştürülüyor.
- Küçültülmüş görüntüdeki yüzlerin konumları ve kodları tespit ediliyor.
- Bulunan yüzler, bilinen yüzlerle karşılaştırılıyor. En iyi eşleşme bulunarak isim atanıyor. Eşleşme yoksa "Bilinmiyor" olarak işaretleniyor.

```
while True:
    ret, frame = video_capture.read()

    if process_this_frame:
        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

        rgb_small_frame = small_frame[:, :, :-1]

        face_locations =
face_recognition.face_locations(rgb_small_frame)
        face_encodings =
face_recognition.face_encodings(rgb_small_frame, face_locations)

        face_names = []
        for face_encoding in face_encodings:
            matches =
face_recognition.compare_faces(known_face_encodings, face_encoding)
            name = "Bilinmiyor"

            face_distances =
face_recognition.face_distance(known_face_encodings, face_encoding)
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
                name = known_face_names[best_match_index]

            face_names.append(name)

        process_this_frame = not process_this_frame
```

Şekil 1.2.1.5. Yüz tanıma için veri tabanı oluşturulması-5

Algılanan yüzlerin konumları, ilk olarak küçültülmüş boyutlarından orijinal boyutlarına yeniden ölçeklenmektedir. Daha sonra, her yüzün etrafına kırmızı renkli dikdörtgenler çizilerek görsel olarak belirginleştirilir. Ayrıca, her yüzün altına, ilgili kişinin ismini veya kodunu gösteren etiketler eklenir. Tüm bu işlemler tamamlandıktan sonra, işlenen video karesi ekranda görüntülenir. Video akışını durdurmak için 'q' tuşuna basılarak döngü sonlandırılabilir. Bu işlem sonucunda, video yakalama süreci durdurulmuş olur ve tüm OpenCV pencereleri kapatılarak program sonlandırılır.

```
for (top, right, bottom, left), name in zip(face_locations,
face_names):

    top *= 4
    right *= 4
    bottom *= 4
    left *= 4

    cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
    cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0,
255), cv2.FILLED)
    font = cv2.FONT_HERSHEY_DUPLEX
    cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255,
255, 255), 1)

    cv2.imshow('Video', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
video_capture.release()
cv2.destroyAllWindows()
```

Şekil 1.2.1.6. Yüz tanıma için veri tabanı oluşturulması-6

Gerçek zamanlı olarak kameradan alınan görüntüler üzerinde yüz algılama ve yüz özelliklerini tespit etmek için Dlib ve OpenCV kütüphanelerini kullanılır. Öncelikle, Dlib kütüphanesi içerisindeki yüz algılayıcı ve 68 nokta yüz hatları tahmin modeli yüklenir. OpenCV ve Dlib kütüphaneleri import edilerek başlanır.

```
import cv2
import dlib
```

Şekil 1.2.1.7. Yüz tanımanın gerçekleştirilmesi

Daha sonra 'detector' ve 'model' değişkenlerinin tanımlaması yapılır. Kamera başlatılır ve 'cap' değişkeni ile kameradan görüntü alınır.

```
detector=dlib.get_frontal_face_detector()
model=dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

cap=cv2.VideoCapture(1)
```

Şekil 1.2.1.8. Yüz tanımanın gerçekleştirilmesi-2

Bir sonsuz döngü içerisinde kamera akışı sürekli olarak alınır ve yüz algılamanın daha etkili olması için gri tonlamaya dönüştürülür. Algılanan yüz 'detector' kullanılarak tespit edilir ve her yüz için yüz noktaları 'model' kullanılarak belirlenir. Algılanan yüz noktalarının her birini ekranda göstermek için küçük mavi daireler çizilir.

```
while True:
    _,frame=cap.read()
    gri=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)

    faces=detector(gri)
    for face in faces:
        points=model(gri,face)
        print(points)

        for i in range(68):
            x,y=(points.part(i).x , points.part(i).y)
            cv2.circle(frame,(x,y),3,(255,0,0),-1)
```

Şekil 1.2.1.9. Yüz tanımanın gerçekleştirilmesi-3

Görüntü, 'cv2.imshow' fonksiyonu ile ekranda gösterilir ve kullanıcı 'q' tuşuna bastığında döngü bitirilir ve program sonlandırılır. Bu işlem 'cv2.waitKey' fonksiyonu ile kontrol edilir. Son olarak, kamera ve tüm pencereler kapatılır.

```
cv2.imshow("frame",frame)

if cv2.waitKey(1) & 0xff==ord("q"):
    break
cap.release()
cv2.destroyAllWindows()
```

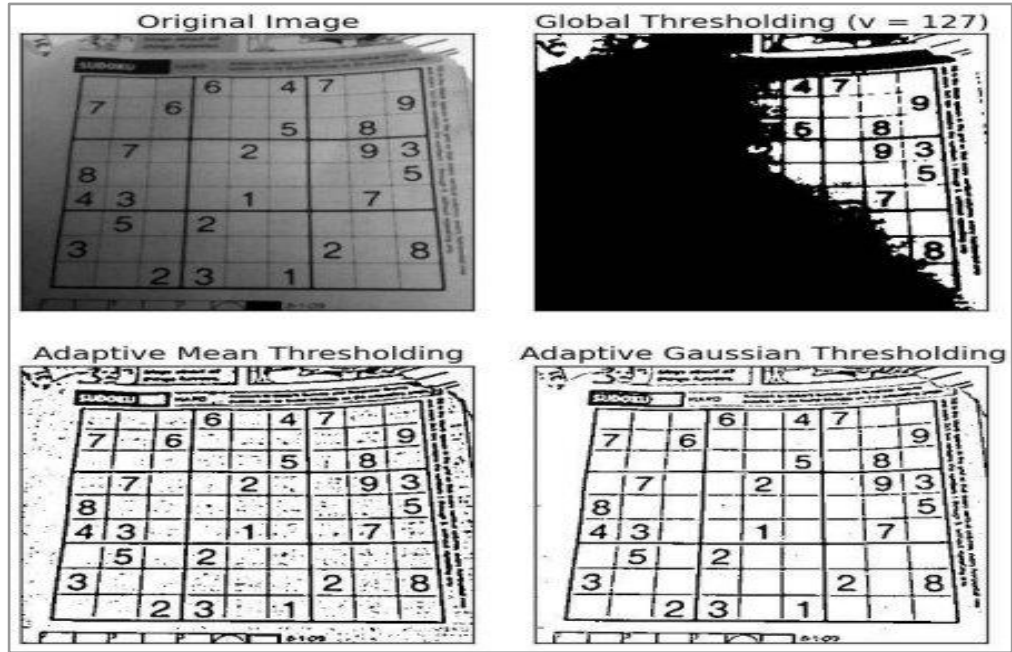
Şekil 1.2.1.10. Yüz tanımanın gerçekleştirilmesi-4

1.2.2. Optik Okuma ve Değerlendirme Modülü

Optik okuma ve değerlendirme, belge üzerindeki işaretleri veya metinleri tarayarak dijital verilere dönüştüren ve bu verileri analiz eden bir süreçtir. Teknolojinin ilerlemesiyle birlikte, optik okuma sistemleri sadece eğitim ve sınav gibi alanlarda değil, aynı zamanda sağlık, finans ve lojistik gibi farklı sektörlerde de yaygın olarak kullanılmaktadır. Bu bölümde, optik okuma ve değerlendirmenin sınav sürecindeki potansiyel uygulaması incelenecektir.

Optik okuma ve değerlendirme, Python programlama dili ve OpenCV ve NumPy kütüphaneleri kullanılarak gerçekleştirilecektir. İşlem aşamaları şunlardır:

- Adayların cevap kâğıtları kamera ile taranarak görüntü dosyaları elde edilecek.
- Algoritma destekli görüntü işleme yazılımı, cevap kâğıdı görüntülerini işleyerek metin ve sayıları algılayacaktır.
- Alınan veriler ile doğrulama ve değerlendirme yapılacaktır.
- Optik okuma sonuçları, aday kimlik bilgileri ve doğru cevap anahtarları blokzincire kaydedilecektir.



Fotoğraf 1.2.2.1. OpenCV kütüphanesinin dokümantasyonundan bir örnek

OpenCV makine öğrenimi ve görüntü işleme için sıklıkla kullanılan bir açık kaynak kodlu kütüphanedir. Kaynaktan görüntünün alınması, boyutlandırılması, yeniden renklendirilmesi veya analiz edilmesi dahil birçok işlevi içerisinde barındırır. **Fotoğraf 1.2.2.1.**' de ise kütüphanenin kendi dokümantasyonundan bir örneği irdelenecektir.

OpenCV içerisindeki Eşikleme (Threshold) fonksiyonunun istediği ilk parametre işlem yapılacak resmin kendisidir. Sonrasındaki parametre ise verilecek eşik değeri. Üçüncü parametre ise bu eşik değeri geçen matris elemanlarının atanmasını istediğimiz değerdir. İlk resimde (1.Resim) eşik değeri 20 olarak belirlendi. 20 üzerindeki bütün değerlere 255 atandı. Kalan değerler ise 0 olarak belirlendi. Bu da yalnızca siyah veya siyaha yakın çok koyu renklerin resimde kalmasını diğer bütün tonların doğrudan beyaz olmasını sağladı.

2 ve 3. resimlerde 80 ve 160 gibi ara değerler verilmiştir. Son olarak 4.resimde ise 200 belirlenmiştir. Burada da 1.resmin aksine beyaz ve beyaza çok yakın açık renkleri 255 olarak atanırken kalan bütün değerler 0 olarak belirlenir. Bu da resmin çok koyu olmasını sağladı. Buradaki değerler, her resim ve her durum için özel olarak belirlenmelidir. Alınan görüntü bozuk olsa bile fonksiyon uygulandıktan sonra görüntü netleşmiş hale gelmektedir. Görüldüğü üzere çizgiler belirginleştirilip okuma daha net yapılır. Threshold fonksiyonunu optik formdaki çizgileri belirginleştirip işaret alanının tespitinde kullanacağız.

Standart optik formun hazırlanma sürecinde, formun net okunabilir ve işlenebilir hale getirilmesi kritik bir adımdır. Bu sebeple standart optik form minimum karışıklığa sebebiyet verecek şekilde sade tasarlanmıştır. Aday, işaretlemelerini bu optik forma yapacak ve sınav sonunda optik formunu kameraya doğru tutacak. Böylelikle kamera hem optik formu hem de adayın yüzünü doğrulamış olacaktır. Daha sonra alınan bu görüntü blokzincir yardımıyla kaydedilecektir. Bu görüntü verileri şifreli bir biçimde saklanarak hem aday cevaplarının güvenliği hem de sınavın güvenliğinin artırılması sağlanacaktır. Python kullanılarak bu sürecin nasıl gerçekleştirileceğine bakalım.

İlk olarak gerekli Python kütüphaneleri import komutuyla içe aktarılmaktadır. OpenCV (cv2) görüntü işleme fonksiyonlarını sağlar, numpy ise sayısal işlemler ve dizilerle çalışmak için kullanılır. Utils kütüphanesi ise harici olarak tanımlanan yardımcı fonksiyonları çağırmak için kullanılır.

```
import cv2
import numpy as np
import utils
```

Şekil 1.2.2.1. Optik okumanın gerçekleştirilmesi

Daha sonra programın çalışması için gerekli bazı temel ayarlar yapılmaktadır. Eğer **‘webCamFeed’** değeri **‘True’** ise, görüntü web kamerasından alınır, aksi takdirde

belirtilen dosya yolundan okunur. Görüntünün boyutları 700x700 olarak belirlenir. Sınav kağıdı her biri 5 seçenekten 5 soru içerir. Doğru cevaplar tanımlanır. Burada cevap kısmında tanımlanan 0 rakamı A seçeneğini, 1 rakamı B seçeneğini, 2 rakamı C, 3 rakamı D ve 4 rakamı da E seçeneğini ifade eder. Yani 1.sorunun cevabı B, 2.sorunun cevabı C, 3.sorunun cevabı A, 4.sorunun cevabı C ve 4.sorunun cevabı E olmaktadır.

```
webCamFeed = True
pathImage = "2.jpeg"
cap = cv2.VideoCapture(1)
cap.set(10,160)
heightImg = 700
widthImg = 700
questions=5
choices=5
ans= [1,2,0,2,4]
```

Şekil 1.2.2.2. Optik okumanın gerçekleştirilmesi-2

While döngüsü ile görüntü sürekli olarak alınır ve işlenir. Görüntü, resize fonksiyonu ile belirlenen boyutlara yeniden boyutlandırılır. Final ve Blank ile gerekirse hata ayıklamanın test edilebilmesi için boş bir görüntü oluşturulur. Ardından Gray ile gri tonlamaya dönüştürülür. Gaussian bulanıklığı pürüzsüzleştirme için uygulanır ve kenarları belirlemek için Canny kenar algılama algoritması kullanılır.

```
while True:

    if webCamFeed:success, img = cap.read()
    else:img = cv2.imread(pathImage)
    img = cv2.resize(img, (widthImg, heightImg))
    imgFinal = img.copy()
    imgBlank = np.zeros((heightImg,widthImg, 3), np.uint8)
    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    imgBlur = cv2.GaussianBlur(imgGray, (5, 5), 1)
    imgCanny = cv2.Canny(imgBlur,10,70)
```

Şekil 1.2.2.3. Optik okumanın gerçekleştirilmesi-3

Bu bölümde amaç tüm sınırların bulunmasıdır. İlk olarak görüntü almak için kopyalar oluşturulur. Daha sonra kenarları belirlenen görüntüdeki tüm konturlar bulunur ve çizilir. Daha sonra dikdörtgen şeklindeki konturlar filtrelenir ve en büyük iki dikdörtgenin köşe noktaları belirlenir. Bu dikdörtgenler sınav kağıdının ve puanlama alanının sınırlarını temsil eder.

```

try:
    imgContours = img.copy()
    imgBigContour = img.copy()
    contours,hierarchy=cv2.findContours(imgCanny,cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
    cv2.drawContours(imgContours, contours, -1, (0, 255, 0), 10)
    rectCon = utils.rectContour(contours)
    biggestPoints = utils.getCornerPoints(rectCon[0])
    gradePoints = utils.getCornerPoints(rectCon[1])

```

Şekil 1.2.2.4. Optik okumanın gerçekleştirilmesi-4

Bir sonraki aşamada, eğer köşe noktaları başarıyla bulunmuşsa, bu noktalar yeniden sıralanır ve perspektif dönüşümü uygulanır. Sırasıyla en büyük dikdörtgenin köşeleri işlenir, tüm köşeler işlenir, en büyük sınır çizilir, noktalar işleme hazırlanır, noktaların yükseklik ve genişlikleri işlenir, noktalar bir matrise çevrilir ve son olarak perspektif çarpıtması uygulanır. Bu işlem sınav kağıdını düzleştirerek daha kolay analiz edilebilir hale getirir.

```

if biggestPoints.size != 0 and gradePoints.size != 0:
    biggestPoints=utils.reorder(biggestPoints)
    cv2.drawContours(imgBigContour, biggestPoints, -1, (0, 255, 0), 20)
    pts1 = np.float32(biggestPoints)
    pts2=np.float32([[0,0],[widthImg,0],[0,heightImg],[widthImg,heightI
mg]])
    matrix = cv2.getPerspectiveTransform(pts1, pts2)
    imgWarpColored=cv2.warpPerspective(img,matrix,(widthImg,heightImg))

```

Şekil 1.2.2.5. Optik okumanın gerçekleştirilmesi-5

Bir üst adımda en büyük dikdörtgene uygulanan sınır çizilmesi, noktaların matrise çevrilmesi gibi işlemler bu kez de ikinci büyük dikdörtgene uygulanarak; kâğıdın çerçeveleri net olarak belirlenmiş oluyor.

```

cv2.drawContours(imgBigContour, gradePoints, -1, (255, 0, 0), 20)
gradePoints = utils.reorder(gradePoints)
ptsG1 = np.float32(gradePoints)
ptsG2 = np.float32([[0, 0], [325, 0], [0, 150], [325, 150]])
matrixG = cv2.getPerspectiveTransform(ptsG1, ptsG2)
imgGradeDisplay = cv2.warpPerspective(img, matrixG, (325, 150))

```

Şekil 1.2.2.6. Optik okumanın gerçekleştirilmesi-6

İşaretlenen cevapları tespit edebilmek için her soru ve seçenek kutusu ayrı ayrı işlenir. Gri tonlamaya dönüştürülen görüntü, eşikleme işlemi ile ikili hale getirilir. Ardından, bu ikili görüntü, her bir soru ve seçenek kutusuna bölünür. Her kutudaki siyah piksel sayısı hesaplanarak bir matrise kaydedilir. Bu sayede aday tarafından işaretlenen seçenekler belirlenir.

```
imgWarpGray = cv2.cvtColor(imgWarpColored,cv2.COLOR_BGR2GRAY)
imgThresh = cv2.threshold(imgWarpGray, 170,
255,cv2.THRESH_BINARY_INV )[1]
boxes = utils.splitBoxes(imgThresh)
countR=0
countC=0
myPixelVal = np.zeros((questions,choices))
for image in boxes:
    totalPixels = cv2.countNonZero(image)
    myPixelVal[countR][countC]= totalPixels
    countC += 1
    if (countC==choices):countC=0;countR +=1
```

Şekil 1.2.2.7. Optik okumanın gerçekleştirilmesi-7

Bu aşamada, her soru için işaretlenen seçenekler belirlenir ve doğru cevaplarla karşılaştırılır. Her soruya verilen cevaplar, en yüksek piksel değerine sahip kutular olarak belirlenir ve doğru cevaplarla karşılaştırılır. Bu karşılaştırma sonucunda, doğru cevapların sayısı hesaplanır ve toplam puan yüzdesi olarak hesaplanır. Ara satırlarda bulunan ve görünümünün test edilmesi amacıyla kullanılan komutlar # simgesi eklenerek deaktif edilmiştir.

```
myIndex=[]
for x in range (0,questions):
    arr = myPixelVal[x]
    myIndexVal = np.where(arr == np.amax(arr))
    myIndex.append(myIndexVal[0][0])
grading=[]
for x in range(0,questions):
    if ans[x] == myIndex[x]:
        grading.append(1)
    else:grading.append(0)
#print("GRADING",grading)
score = (sum(grading)/questions)*100 # FINAL NOTU
#print("SCORE",score)
```

Şekil 1.2.2.8. Optik okumanın gerçekleştirilmesi-8

Artık cevapları ve sonuçları görüntüleyebileceğimiz bu bölümde, görüntü işleme adımlarının sonucunda elde edilen işaretli cevaplar, `utils.showAnswers()` fonksiyonu aracılığıyla görselleştirilir. Bu aşamada, işaretli cevapların üzerine bir ızgara çizilmesi için `utils.drawGrid()` fonksiyonu kullanılır. Ardından, işlenmiş görüntülerin boyutlarına uygun yeni bir boş görüntü oluşturulur ve `utils.showAnswers()` fonksiyonu yeniden çağrılarak bu boş görüntüye işaretli cevaplar çizilir. Sonrasında, perspektif dönüşüm matrisi alınarak işaretli cevaplar ters perspektif dönüşümü uygulanarak orijinal konumlarına getirilir.

```
utils.showAnswers(imgWarpColored,myIndex,grading,ans)
utils.drawGrid(imgWarpColored)
imgRawDrawings = np.zeros_like(imgWarpColored)
utils.showAnswers(imgRawDrawings, myIndex, grading, ans)
invMatrix = cv2.getPerspectiveTransform(pts2, pts1)
imgInvWarp = cv2.warpPerspective(imgRawDrawings, invMatrix,
(widthImg, heightImg))
```

Şekil 1.2.2.9. Optik okumanın gerçekleştirilmesi-9

Sonuçların görselleştirilmesi için, elde edilen notun eklendiği bir sonuç alanı oluşturulur. Bu sonuç alanına metin olarak notun yüzdesi eklenir ve görüntüye `cv2.putText()` fonksiyonu kullanılarak yerleştirilir. Daha sonra, cevaplar ve sonuçlar birleştirilerek toplu bir sonuç görüntüsü oluşturulur. Oluşturulan bu sonuç görüntüsü, `cv2.imshow()` fonksiyonu aracılığıyla kullanıcıya gösterilir.

```
imgRawGrade = np.zeros_like(imgGradeDisplay,np.uint8)
cv2.putText(imgRawGrade,str(int(score))+"%",(100,100),cv2.FONT_HERSHEY_
HEY_TRIPLEX,4,(255,0,0),10)

invMatrixG = cv2.getPerspectiveTransform(ptsG2, ptsG1)
imgInvGradeDisplay = cv2.warpPerspective(imgRawGrade, invMatrixG,
(widthImg, heightImg))

imgFinal = cv2.addWeighted(imgFinal, 1, imgInvWarp, 1,0)
imgFinal = cv2.addWeighted(imgFinal, 1, imgInvGradeDisplay, 1,0)

imageArray = ([img,imgGray,imgCanny,imgContours],
[imgBigContour,imgThresh,imgWarpColored,imgFinal])
cv2.imshow("Final Sonucu", imgFinal)
```

Şekil 1.2.2.10. Optik okumanın gerçekleştirilmesi-10

Herhangi bir hata durumunda, işlemlerin gerçekleştirilmesi için boş bir görüntü oluşturulur ve sonuçlar görselleştirilir. Son olarak, işlemler sırasında elde edilen tüm görüntülerin başlıkları belirlenir ve birleştirilerek ekrana yansıtılır.

```
except:
    imageArray = ([img,imgGray,imgCanny,imgContours],
                  [imgBlank, imgBlank, imgBlank, imgBlank])
lables =
[["Orjinal","GriTonlar","SiyahBeyazKenarlar","SINIRCizgileri"],
 ["DikdortgenKoseleri","SiyahBeyazCerçeve","RenkliCerçeve","Final"]]
```

Şekil 1.2.2.11. Optik okumanın gerçekleştirilmesi-11

Burada stackImages fonksiyonu kullanılarak imageArray içindeki görüntülerin yatay ve dikey olarak birleştirilmesiyle oluşturulur. Bu işlem, görüntülerin belirli bir ölçekte ve etiketlerle birleştirilmesini sağlar. Sonuç olarak elde edilen birleştirilmiş görüntü, "Sonuc" başlığı altında cv2.imshow() fonksiyonuyla kullanıcıya gösterilir.

```
stackedImage = utils.stackImages(imageArray,0.55,lables)
cv2.imshow("Sonuc",stackedImage)
```

Şekil 1.2.2.12. Optik okumanın gerçekleştirilmesi-12

"S" tuşuna basıldığında, cv2.imwrite() fonksiyonu ile stackedImage görüntüsü belirtilen yol ve adrese kaydedilir. Ardından, cv2.rectangle() ve cv2.putText() fonksiyonları kullanılarak kayıt işleminin başarıyla gerçekleştiği bildirimi stackedImage üzerine çizilir ve cv2.imshow() fonksiyonu ile kullanıcıya gösterilir. Son olarak, klavyeden "Q" tuşuna basılması durumunda döngü sonlandırılır ve program kapatılır.

```
if cv2.waitKey(1) & 0xFF == ord('s'):
    cv2.imwrite(r'C:Scanned\myImage\Final'+str(count)+".jpeg",imgFinal)

    cv2.rectangle(stackedImage, ((int(stackedImage.shape[1] / 2) -
    230), int(stackedImage.shape[0] / 2) + 50), (1100, 350), (255, 255,
    0), cv2.FILLED)
    cv2.putText(stackedImage,"Kaydedildi!", (int(stackedImage.shape[1] /
    2) - 200, int(stackedImage.shape[0] / 2)),cv2.FONT_HERSHEY_TRIPLEX,
    3, (0, 0, 255), 5, cv2.LINE_AA)
    cv2.imshow('Sonuc', stackedImage)
    cv2.waitKey(300)
    count += 1
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

Şekil 1.2.2.13. Optik okumanın gerçekleştirilmesi-13

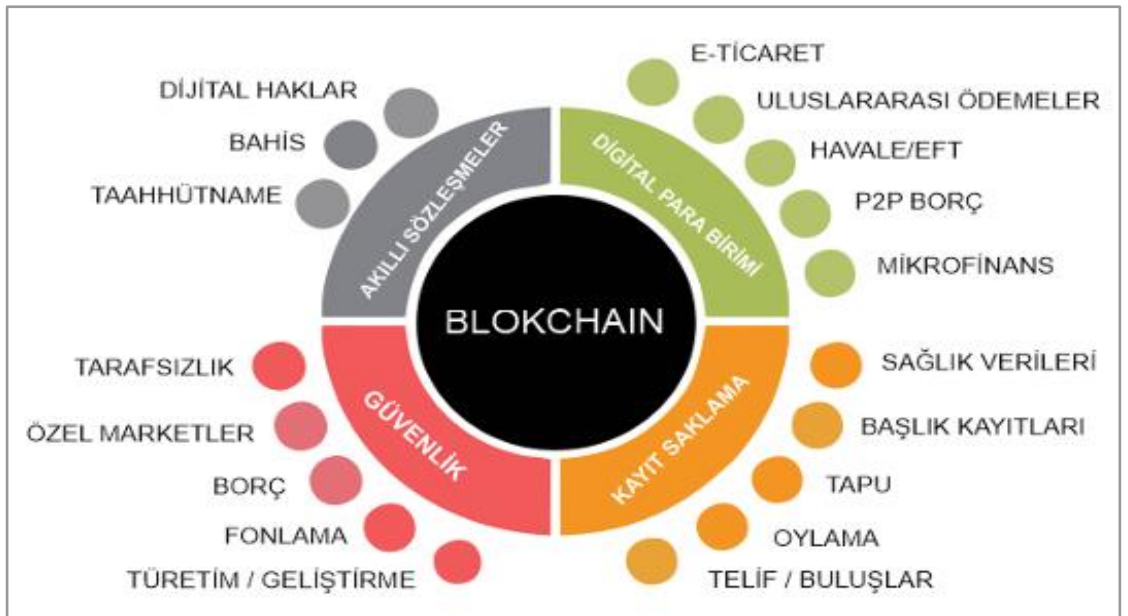
İKİNCİ BÖLÜM

BLOKZİNCİR

2.1. Blokzincir Alanında Yapılan Çalışmaların Analizi

Blokzincir kavramı, modern teknoloji dünyasında görece yeni bir olgu olarak karşımıza çıkmakta ve üzerine yapılan akademik çalışmalar henüz sınırlı sayıdadır. Ancak, finans sektörü dışında hızla yaygınlaşan kullanım alanları, blokzincir teknolojisinin popülerliğini ve potansiyelini gözler önüne sermektedir. Bu durum, özellikle farklı sektörlerde yenilikçi uygulamaların geliştirilmesi ve entegrasyonu açısından büyük bir ilgi uyandırmakta ve teknolojinin çok yönlü faydalarını ön plana çıkarmaktadır. Blokzincirin sunduğu bu yenilikçi fırsatlar, teknolojinin temel özellikleri ve avantajları ile doğrudan ilişkilidir.

Değiştirilemezlik, veri izlenebilirliği ve güvenlik gibi özellikleri ve merkezi olmayan doğası, başarısını garantileyen başlıca itici faktörler olmuştur. Blokzincir teknolojisinin avantajlarına rağmen, ölçeklenebilirlik, enerji tüketimi ve altyapı gereksinimleri gibi birkaç zorluk vardır. Bulutun blokzincir ile entegrasyonunun bu zorlukları azaltabileceğine ve merkezi olmayan uygulamaların geliştirilmesini ve dağıtımını artırabileceğine inanılmaktadır [23]. Blokzincirin yaygın kullanım alanlarından **Çizelge 1.1.**'de bahsedilmiştir. **Fotoğraf 2.1.1.**'de ise şematize edilmiş hali görülmektedir.



Fotoğraf 2.1.1. Blokzincir kullanım alanları [24].

Bu bölümde ise blokzincir alanında yapılan çalışmaların, literatüre ve bu tez çalışmasına katkıları açısından incelenecektir. Blokzincir teknolojisi üzerine yapılan akademik çalışmalar, blokzincirin çeşitli alanlarda nasıl kullanılabileceğine dair önemli bilgiler sunmaktadır.

Blokzincir alanında gerçekleştirilen araştırmaları derleyen bir inceleme makalesi niteliğindeki çalışma, blokzincir alanındaki mevcut literatürü sistematik bir şekilde ele alarak, konuyla ilgili yapılan çalışmaların genel bir değerlendirmesini sunmaktadır [24]. Bu gibi inceleme makaleleri, akademik alanda yapılan araştırmaların kapsamını ve eğilimlerini anlamak için önemli bir kaynak oluşturmaktadır.

IoT cihazlarının güvenliği ve gizliliği üzerine yapılan bir çalışmada, blokzincir teknolojisinin bu alanda nasıl uygulanabileceğini incelenmiştir. IoT uygulamaları, kullanıcıları gizlilik tehditlerine maruz bırakabileceği için, bu tür uygulamalarda gizlilik gereksinimlerinin ele alınması ve tasarımdan itibaren gizlilik sağlayacak çözümlerin geliştirilmesi kritik öneme sahiptir. Araştırma, blokzincir teknolojisinin anonimlik özelliklerini kullanarak, kullanıcıların ve verilerinin gizliliğini artırmayı amaçlayan bir mimari önermektedir. Akıllı araçlar için bir IoT uygulamasında bu mimarinin uygulanabilirliği deneysel olarak değerlendirilmiş, başarılı sonuçlar elde edilmiştir [25].

Akıllı sağlık izleme sistemlerinin ayrılmaz bir parçası haline gelen Tıbbi Nesnelerin İnterneti (IoMT) üzerine yürütülen bir başka çalışmada, IPFS ve blokzincir teknolojilerinin entegrasyonunu ele almışlardır. IoMT, acil durum ve hastane hizmetlerinin yanı sıra sağlıklı bir yaşam tarzını sürdürmek için gerçek zamanlı hasta verilerini uzaktan izlemeye olanak sağlar. Endüstri 5.0 ve 5/6G teknolojileri sayesinde, geniş bir yelpazede insan biyolojik verilerini toplayan ve kablosuz ağlar üzerinden gerçek zamanlı olarak ileten maliyet etkin sensörler ve cihazlar geliştirilmiştir. Her gün büyük miktarda hasta ve cihaz kaydı yapılan IoMT ağında, büyük veri veya sağlık verisi üretimi gerçekleşmektedir. Bu verilerin gizliliği ve güvenliği, olası suistimalleri önlemek adına korunmalıdır. Bu amaçla, blokzincir ve gezegenler arası dosya sistemi (IPFS) ile entegre edilmiş üç seviyeli bir ağ önerilmiştir. Sunulan çerçeve, ağın ölçeklenebilirliğini doğrulamak için çeşitli ağ aktiviteleri açısından değerlendirilmiş ve karmaşık verileri işleme konusunda verimli olduğu bulunmuştur [26].

Tek bir sunucuda aşırı miktarda veriyi yönetmenin zorluğunu ve dinamik bir ortamda düzensiz veriler için güvenlik önlemlerinin önemini vurgulandığı bir çalışmada,

Mobil Uç Bilişim (Mobile Edge Computing) tabanlı bir dağıtık ortamda savunmasız veri güvenlik risklerine dikkat çekmektedirler. Uç bilişim tabanlı bir ortamda eğitim için uygun bir perspektif sunulduğu belirtilmektedir. Blokzincirin Konsensüs yaklaşımının ve makine öğrenimi tekniklerinin, uç bilişim ile birleştirilmesi, veri güvenliğini artırabilir, açığa çıkan veri olasılığını azaltabilir ve veri ihlali riskini düşürebilir. Önerilen model, blokzincirin Konsensüs yaklaşımı ve makine öğrenimi teknikleri kullanılarak veri güvenliğini içermektedir. Ayrıca, önerilen algoritmalar bir kenar bilişim ortamında uygulanmış ve istemci gizliliği blokzincir sunucuları kullanılarak korunmuştur. Sonuç olarak, bir uç bilişim tabanlı ortamda Blokzincir Tabanlı Bir Eğitim Modelinin uygulanabilirliği gösterilmiştir [27].

İki taraf arasında kimlik doğrulaması yapmak için BAN (Burrows-Abadi-Needham) mantığını kullanarak yapılan bir çalışma, tıbbi bilgi depolamanın hem değiştirilemezliğini hem de güvenliğini garanti etmektedir. Çalışma, bilgilerin yetkisiz erişime karşı korunmasını ve verilerin bütünlüğünün korunmasını sağlayıp, tıbbi bilgi yönetiminde blokzincir teknolojisinin potansiyelini vurgulayarak, güvenli ve güvenilir veri depolama çözümleri sunmaktadır [28].

Blokzincir teknolojisine dayalı siber güvenlik sistemlerinin geliştirilmesine yönelik çözüm önerileri sunan bir çalışmada hem hükümet hem de özel sektör için uygulanabilir güvenlik çözümleri üzerine odaklanılmıştır. Blokzincir teknolojisinin güvenlik altyapısını güçlendirme potansiyeli, bu sistemlerin daha güvenilir ve verimli hale gelmesini sağlamaktadır [29].

Yukarıda incelenen çalışmalar, blokzincir teknolojisinin geniş bir yelpazede sunduğu faydaları ve potansiyel uygulamalarını göstermektedir. Blokzincirin güvenlik, gizlilik, veri bütünlüğü ve ölçeklenebilirlik gibi önemli konularda sunduğu çözümler, çeşitli endüstrilerde bu teknolojinin benimsenmesine yönelik ilginin artmasına neden olmaktadır. Bu tez çalışmasında örnek uygulamalar genişletilerek ve ihtiyaca göre değiştirilecek kullanılacaktır.

2.2. Blokzincir Modülleri

Blokzincir modülleri, bu tez çalışmasında çok yönlü bir güvenlik ve doğruluk katmanı olarak işlev görmektedir. Veri güvenliği ve değiştirilemezlik açısından, blokzincir verilerin dağıtık bir defterde saklanmasını sağlar. Bu yapı, verilerin değiştirilmesini veya silinmesini son derece zor hale getirir. Çalışmada, sınav sonuçları

ve adayların yüz tanıma verileri blokzincire kaydedildiğinde, veriler kriptografik olarak güvence altına alınır ve herhangi bir değişiklik yapılamaz hale gelir. Böylece, art niyetli kişilerin sınav sonuçlarını değiştirme girişimleri önlenir. Blokzincir, her bir veri girişi için dijital bir imza oluşturur ve bu imzalar zincirleme bir yapıda birbirine bağlı olduğundan verilerin bütünlüğünü korunur.

Her bir işlem blokzincirde kronolojik olarak saklanır ve bu işlemler geriye dönük olarak izlenebilir. Sınav yetkililerinin ve denetçilerin, sınav süreci boyunca herhangi bir müdahale veya değişiklik olup olmadığını kolayca kontrol etmelerine olanak tanır. Sistem, sınav sonuçlarının nasıl ve ne zaman kaydedildiği hakkında tam şeffaflık sağlar.

Blokzincir hassas verilerin güvenliğini sağlamak için şifreleme teknikleri kullanır. Bu çalışmada, her adayın sınav sonucu ve yüz tanıma verisi, yalnızca yetkili kişiler tarafından görüntülenebilir şekilde şifrelenecektir. Bu sayede, yetkisiz kişilerin verilere erişim engellenir. Aynı zamanda, blokzincir üzerinde saklanan verilerin erişim kayıtları tutulur ve kimlerin hangi verilere eriştiği denetlenebilir.

Merkeziyetsizlik yönünden, blokzincir verilerin merkezi bir sunucuda değil, dağıtık bir ağ üzerinde saklanmasını sağlar. Sistemin tek bir hata noktasına sahip olmasını engeller ve veri kaybı riskini minimize eder. Sınav sonuçları ve ilgili veriler, birden fazla düğümde saklanarak, veri kaybı veya sistem arızaları durumunda bile erişilebilir ve güvenli kalır.

Blokzincir modüllerinin kullanımı, sınav sürecine olan güveni artıracaktır. Adaylar, sınav sonuçlarının güvenli bir şekilde saklandığını ve değiştirilmediğini bilerek, sınav sistemine daha fazla güven duyarlar. Aynı zamanda, sınav yetkilileri ve denetçiler de verilerin doğruluğunu ve bütünlüğünü kolayca teyit edebilirler. Sınavların şeffaflığına ve güvenilirliğine önemli bir katkı sağlanmış olur.

Bu tez çalışmasında blokzincir teknolojisinin kullanılması, sınav sonuçlarının ve aday verilerinin güvenli, şeffaf ve izlenebilir bir şekilde saklanmasını sağlayarak hem adayların hem de sınav yetkililerinin sisteme olan güvenini artıracaktır.

2.2.1. Kullanıcı Dostu Arayüzün Tasarlanması

Bu çalışmada kullanıcı dostu bir arayüz, sınav sürecine katılan tüm paydaşların sistemi rahatça kullanabilmesini sağladığı için son derece önemlidir. Adaylar, sınav sonrası sonuçlarını ve yüz tanıma verilerini kolayca görüntüleyebilmekte, sınav süreçleri hakkında net bilgi sahibi olmaktadır. Karmaşık bir arayüz, adayların etkin kullanımını

zorlaştırarak sınav sonuçlarına erişimlerinde sorunlara yol açabilir. Sınav yetkilileri ve denetçiler ise verileri hızlı ve doğru bir şekilde inceleyebilmeli, analiz edebilmeli ve böylece olası hataları ve gecikmeleri minimize edebilmelidir. Kolay öğrenilen ve kullanılan bir arayüz, eğitim ve teknik destek ihtiyaçlarını azaltarak maliyet tasarrufu sağlar ve sistemi daha verimli hale getirir. Aynı zamanda, yanlış veri girişi veya yanlış bilgilere erişim gibi hataları önleyerek güvenliği ve gizliliği artırır. Böylelikle genel kullanıcı memnuniyeti artarak, sisteme ve sınav süreçlerine olan güven pekişir ve sistemin kabulü ve yaygın kullanımı teşvik edilir. Tüm bunlar ise **‘Görüntü işleme ve blokzinciri tabanlı bir iyileştirilmiş optik okuma ve sınav değerlendirme sistemi’** nin başarısını doğrudan etkiler.

Geliştirilecek olan sistem, Python programlama dilini temel almakla birlikte, kullanıcı arayüzü için HTML (HyperText Markup Language) ve CSS (Cascading Style Sheet) metin işaretleme dillerinden yoğun bir şekilde faydalanarak hem kullanıcı deneyimini zenginleştirmeyi hem de web tabanlı bir arayüz sunmayı hedeflemektedir.

Web uygulamaları oluşturmak için kullanılan bir Python kütüphanesi olan Streamlit, kullanıcı arayüzü oluşturmak ve Python kodunu web uygulamasına dönüştürmek için kullanılacak ve böylece kullanıcılar web üzerinden uygulamaya erişebilecekler.

Blokzincir uygulamaları geliştirmek için kullanılan bir Python kütüphanesi olan Web3.py, Ethereum blokzincir ile etkileşime geçmek, akıllı sözleşmelerle çalışmak ve blokzincir verilerini işlemek için kullanılacak. Bu sayede, güvenli ve merkeziyetsiz veri saklama ve işleme işlemleri gerçekleştirilebilecek.

Python'un gücünden ve çeşitli kütüphanelerinden faydalanarak hem güçlü bir arka plan işleme motoru sağlanacak hem de HTML/CSS ile estetik ve kullanıcı dostu bir arayüz oluşturulacaktır. Sonuç olarak, yazılımın kullanıcı odaklılık ve işlevselliği ön planda tutularak geliştirilmesi hedeflenmektedir.

Python programlama dilinde oluşturulan algoritmanın başlangıcında gerekli kütüphaneler ve modüller içe aktarılmıştır. Streamlit kütüphanesi, kullanıcı arayüzünü oluşturmak için kullanılırken, **‘sleep’** fonksiyonu, giriş işlemi sonrası kısa bir bekleme süresi sağlamak için kullanılmıştır. Ayrıca **‘make_sidebar’** fonksiyonu, uygulamanın yan menüsünü oluşturmak amacıyla **‘navigation’** modülünden çağırılmıştır. Bu modül kullanıcıların uygulama içerisinde kolayca gezinebilmelerini sağlamaktadır.

```
import streamlit as st
from time import sleep
from navigation import make_sidebar

make_sidebar()
```

Şekil 2.2.1.1. Kullanıcı dostu arayüzün tasarlanması

Daha sonra, ‘st.title’ fonksiyonu ile sayfanın başlığı "Yüksek Lisans Tez Çalışması Sınavı" olarak belirlenmiştir. Kullanıcılara giriş yapmaları için bilgilendirme mesajı ‘st.write’ fonksiyonu ile gösterilmiştir. Kullanıcıdan giriş bilgilerini almak için ‘st.text_input("Aday ID")’ ve ‘st.text_input("Şifre",type="password")’ fonksiyonu kullanılmıştır.

```
st.title("Yüksek Lisans Tez Çalışması Sınavı")

st.write("Giriş yapmak için aşağıdan Aday ID ve şifrenizi giriniz  
(Deneme ID `test`, Deneme Şifre `test`)." )

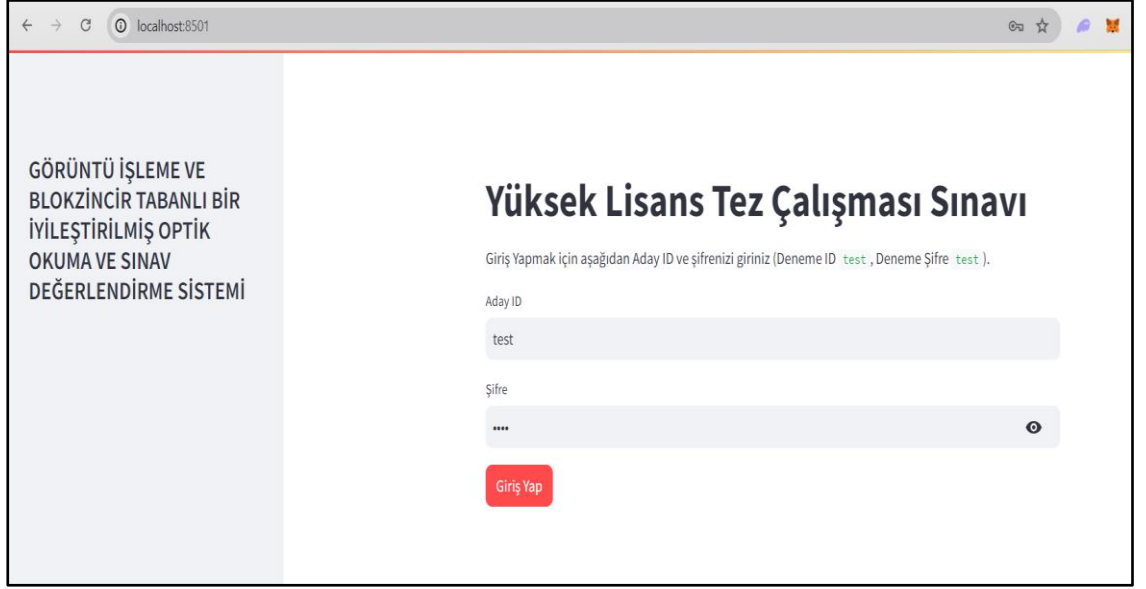
username = st.text_input("Aday ID")
password = st.text_input("Şifre", type="password")
```

Şekil 2.2.1.2. Kullanıcı dostu arayüzün tasarlanması-2

Kullanıcının giriş yapması için ‘st.button("Giriş Yap", type="primary")’ fonksiyonu ile bir buton oluşturulmuştur. Tasarlanan uygulamanın modüllerini test edebilmek için ‘Aday ID =test’ ve ‘Şifre=test’ olarak tanımlama yapılmıştır. Uygulanacak olan sınav sisteminde her kullanıcının kendi Aday ID ve şifresi olacaktır. Test sistemine Aday ID ve şifre doğru olarak girildiğinde giriş işlemi başarılı kabul edilerek kullanıcıya ‘st.success’ fonksiyonu ile başarı mesajı gösterilmektedir. Daha sonra ‘sleep(0.5)’ fonksiyonu ile kısa bir bekleme süresinin ardından, kullanıcı ‘st.switch_page("pages/page1.py")’ ifadesi ile ilgili sayfaya yönlendirilmektedir.

```
if st.button("Giriş Yap", type="primary"):
    if username == "test" and password == "test":
        st.session_state.logged_in = True
        st.success("Giriş İşlemi Başarılı")
        sleep(0.5)
        st.switch_page("pages/page1.py")
```

Şekil 2.2.1.3. Kullanıcı dostu arayüzün tasarlanması-3

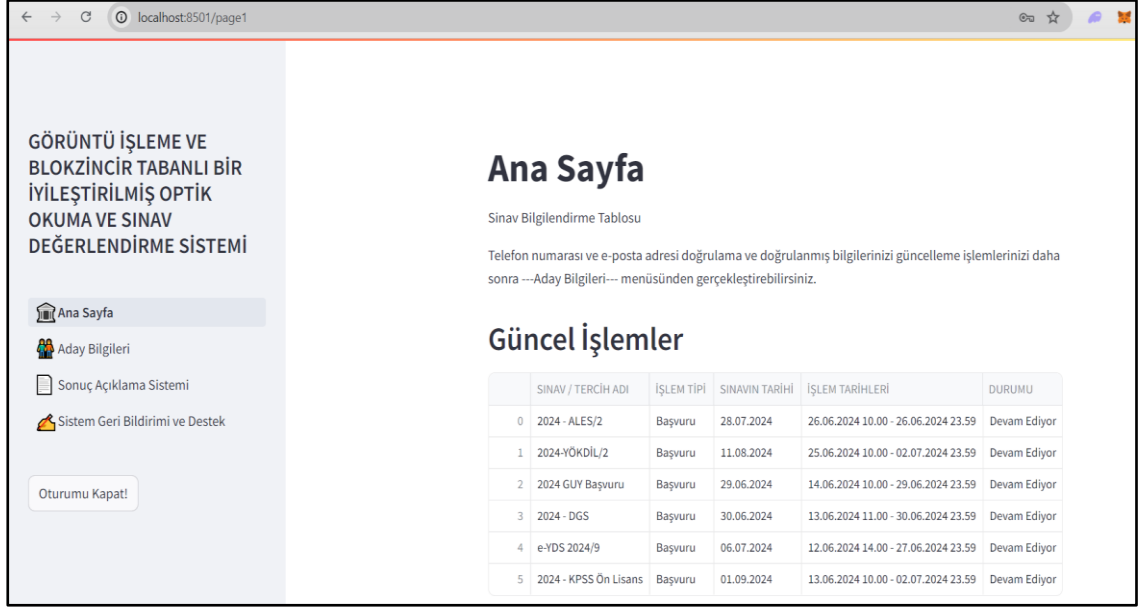


Fotoğraf 2.2.1.1. Arayüz giriş sayfası

Giriş yapıldıktan sonra sol tarafta oluşturulan navigasyon menüsü görüntülenir. Navigasyon menüsünde ilgili fonksiyonlar kullanılarak 4 adet sayfa tanımlanmıştır. **‘with’** bloğu içinde **‘st.sidebar’** kullanılarak kenar çubuğu oluşturulur. Ayrıca **‘st.title’** ile kenar çubuğunda büyük bir başlık eklenir. **‘st.write(“”)’** komutu ile bağlantılar arasında ve kenar çubuğunun altında boş bir alan bırakmak için ekstra boş satırlar eklenir. Son olarak **‘st.button’** fonksiyonu, "Oturumu Kapat!" adlı bir buton oluşturur. Kullanıcı bu butona tıkladığında, **‘if’** bloğu içindeki kod çalıştırılır. **‘st.button’** fonksiyonu, butona basıldığında True döner ve sistemden çıkış yapılır.

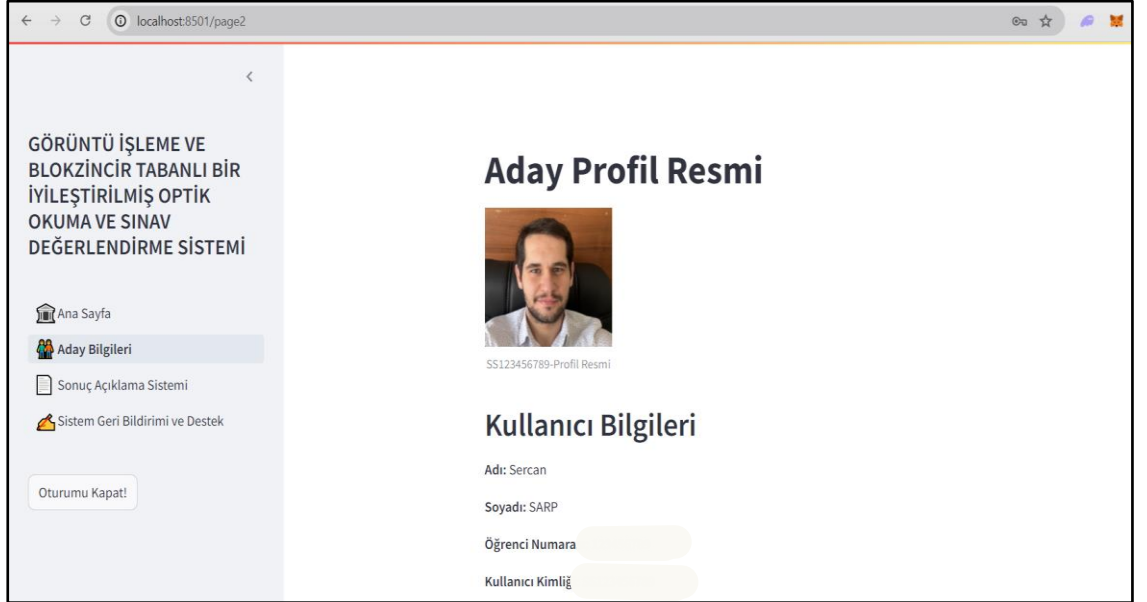
```
def make_sidebar():
    with st.sidebar:
        st.title("GÖRÜNTÜ İŞLEME VE BLOKZİNCİR TABANLI BİR İYİLEŞTİRİLMİŞ OPTİK OKUMA VE SINAV DEĞERLENDİRME SİSTEMİ")
        st.write("")
        if st.session_state.get("logged_in", False):
            st.page_link("pages/page1.py", label="Ana Sayfa")
            st.page_link("pages/page2.py", label="Aday Bilgileri")
            st.page_link("pages/page3.py", label="Sonuç Açıklama")
            st.page_link("pages/page4.py", label="Sistem Destek")
            st.write("")
        if st.button("Oturumu Kapat!"):
            logout()
```

Şekil 2.2.1.4. Kullanıcı dostu arayüzün tasarlanması-4



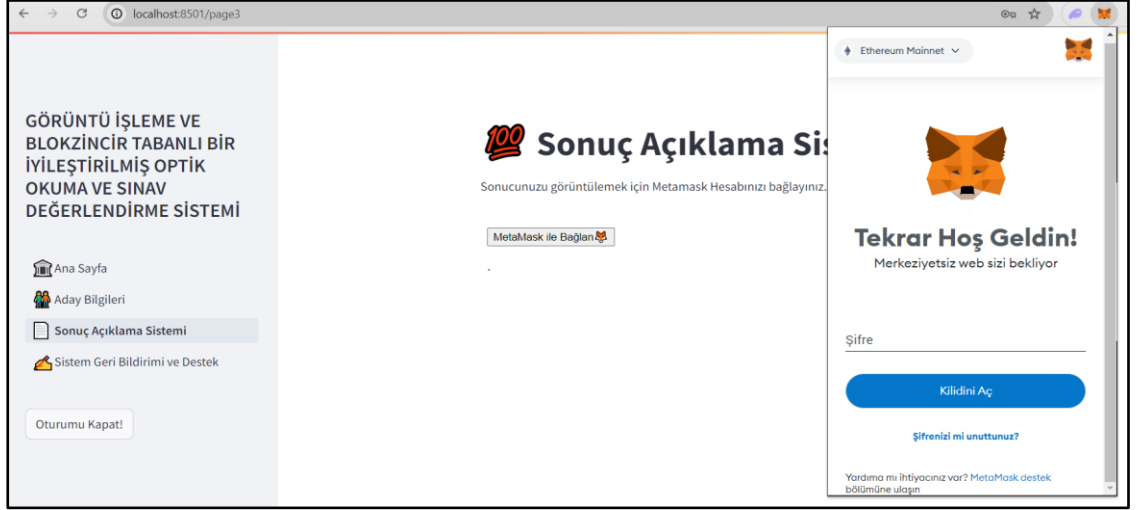
Fotoğraf 2.2.1.2. Ana sayfa ekranı

Arayüzdeki açılış sayfası olan ‘Ana Sayfa’ gibi ‘Aday Bilgileri’, ‘Sonuç Açıklama Sistemi’ ve ‘Sistem Geri Bildirimi ve Destek’ sayfaları da bulunmaktadır. Aşağıda bu sayfaların her birinin içerdiği bilgiler ve örnek ekran görüntüleri yer almaktadır.



Fotoğraf 2.2.1.3. Aday bilgileri ekranı

Aday bilgileri sayfasındaki bilgiler olduğu temel kimlik bilgilerini içermektedir. Profil resmi, kullanıcının görsel olarak tanımlanmasını sağlar, öğrenci numarası ise kullanıcıyı kurum içinde benzersiz olarak tanımlayan bir numaradır. Kullanıcı kimliği, sistem içi işlemlerde ve veritabanı kayıtlarında referans olarak kullanılabilir.



Fotoğraf 2.2.1.4. Sonuç açıklama ekranı

Sonuç açıklama bölümü sınav sonuçlarının güvenli ve şeffaf bir şekilde görüntülenmesi amacıyla blockchain teknolojisi kullanılarak tasarlanmıştır. Kullanıcıların sınav sonuçlarını görüntüleyebilmeleri için Metamask hesaplarını bağlamaları gerekmektedir. Metamask, blockchain tabanlı uygulamalarda kullanıcıların kimlik doğrulamasını ve veri güvenliğini sağlar. Böylece sınav sonuçlarının manipüle edilmesinin veya yetkisiz erişimlerin önüne geçilir.



Fotoğraf 2.2.1.5. Sistem geri bildirimi ve destek ekranı

Geri bildirim ve destek sayfası, kullanıcıların sistemle ilgili düşüncelerini, önerilerini ve karşılaştıkları sorunları doğrudan iletebilmeleri için tasarlanmıştır. Bu formun etkin kullanımı, sistem performansını ve kullanıcı deneyimini geliştirmek amacıyla kritik geri bildirimlerin toplanmasını sağlar.

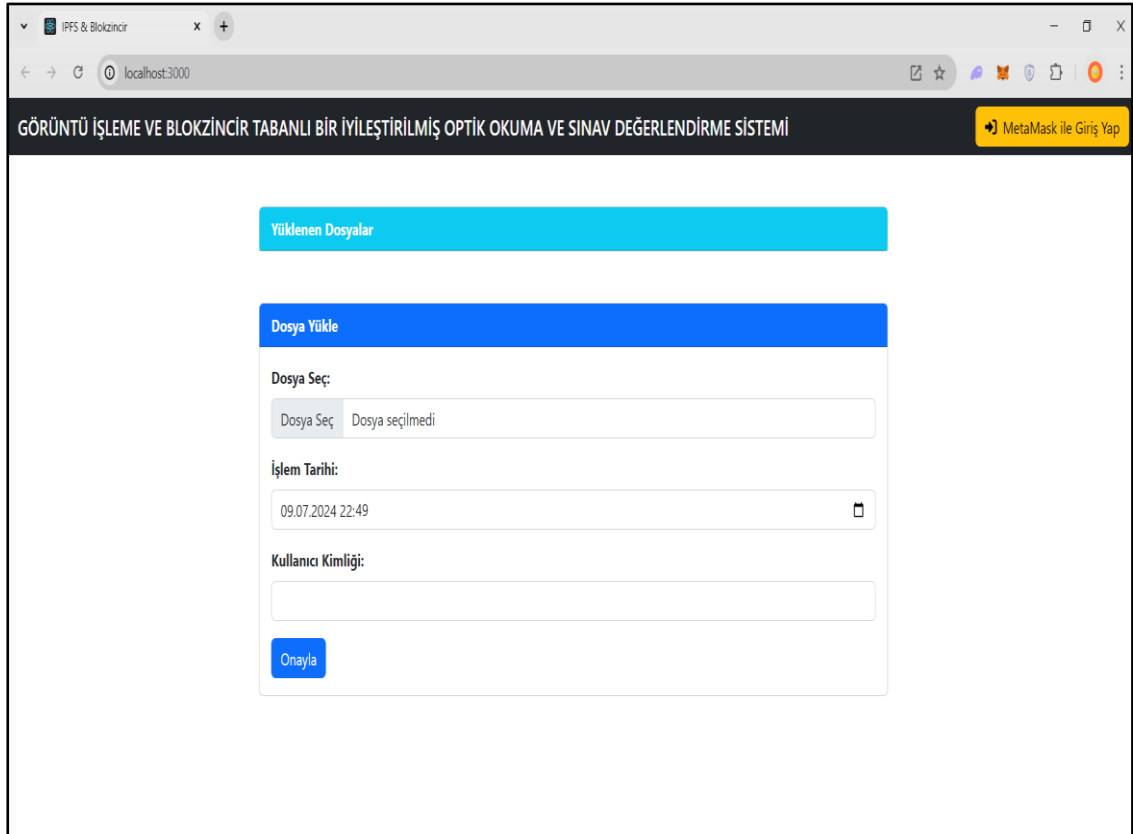
2.2.2. Görüntü Verilerinin Blokzincirde Depolanması

Blokzincir teknolojisi, merkeziyetsiz yapısı ve yüksek güvenlik özellikleri ile bu tür sistemlerde güvenliği ve veri bütünlüğünü sağlamak için ideal bir çözüm sunmaktadır. Verilerin merkezi bir otoriteye ihtiyaç duymadan güvenli bir şekilde saklanmasına ve işlenmesine olanak tanıyan bir teknoloji olan blokzincir, eğitim alanında da sınav sonuçlarının güvenilirliği ve şeffaflığı açısından önemli avantajlar sağlayacaktır. Tez çalışmasının bu aşamasında, IPFS kullanılarak görüntü verilerinin blokzincir üzerinde depolanması sağlanacaktır. IPFS, merkezi olmayan ve dağıtık bir dosya sistemi olarak, verilerin blokzincir üzerinde saklanması sırasında oluşabilecek veri yoğunluğunu ve maliyetini azaltarak, daha verimli bir depolama çözümü sunmaktadır. React tabanlı geliştirilen uygulama ile görüntü dosyalarının yüklenmesi, akabinde bu dosyaların IPFS ve blokzincir üzerinde güvenli bir şekilde saklanması sağlanır. MetaMask aracılığıyla sisteme giriş yapılarak dosyalar yüklenebilir ve bu dosyaların doğrulanabilirliğini sağlanabilir.

Tez çalışmasının bu bölümünde JavaScript kullanılmasının birkaç önemli nedeni vardır. Öncelikle, JavaScript modern web geliştirme için vazgeçilmez bir dil olup, web tarayıcıları tarafından doğal olarak desteklenir. Çalışma, dosya yükleme, dosya önizleme ve kripto cüzdanlarına bağlanma gibi etkileşimli işlemleri gerektirdiğinden; JavaScript, güçlü kütüphaneleri ve framework' leri (örneğin React) sayesinde bu tür etkileşimli arayüzlerin kolayca geliştirilmesine olanak tanır. Ayrıca, JavaScript'in Web3.js gibi kütüphaneleri, Ethereum ve diğer blokzincir ağlarıyla etkileşimi kolaylaştırır. JavaScript'in geniş bir ekosistemi ve büyük bir geliştirici topluluğu bulunmaktadır. Buda, geliştirme sürecinde karşılaşılan sorunların hızlı bir şekilde çözülmesini ve yeni özelliklerin eklenmesini kolaylaştırır. JavaScript kullanımı bu çalışmada hem kullanıcı deneyimini geliştirmek hem blokzincir ve IPFS entegrasyonlarını kolaylaştırmak, hem de geniş topluluk desteğinden yararlanmak için en uygun seçenektir. Sözü edilen bu avantajlardan yararlanılarak kullanıcı dostu bir uygulama oluşturulmuştur.

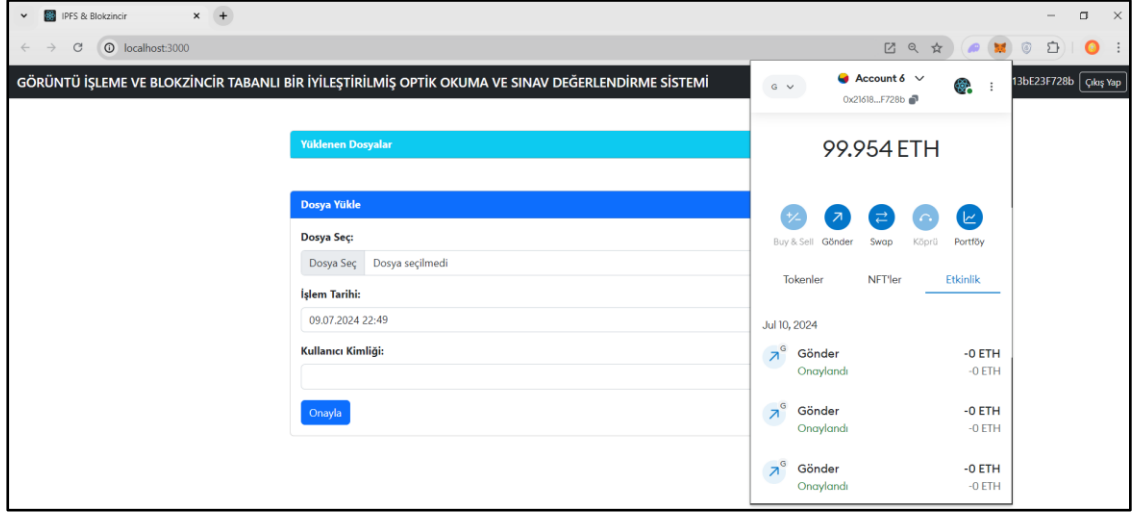
Uygulamanın ana bileşeni olan '**app.js**' dosyası, kullanıcı arayüzünü ve tüm işlevleri yönetir. İlk olarak MetaMask bağlantısını kurmak için '**loadWeb3**' ve mevcut tarih-saat bilgisini almak için '**setCurrentDateTime**' metotları çalıştırılır. MetaMask bağlantısı kurulduğunda, kullanıcı hesapları ve ağ bilgileri alınarak akıllı sözleşme yüklenir. Ayrıca, GMT +3 saat dilimine göre mevcut tarih ve saat bilgisi de ayarlanır.

Dosya yüklenmek istendiğinde, **'captureFile'** metodu dosyayı okuyup **'buffer'** ve **'fileType'** durumlarını günceller. Bu sayede, yüklenen dosyanın türü ve içeriği saklanır. Yüklenen dosyanın önizlemesi **'renderFilePreview'** metodu ile yapılır. Bu metod, dosya türüne göre uygun bir önizleme sunar; örneğin, bir görüntü dosyası için küçük bir önizleme resmi gösterilir. Dosya yükleme formu doldurulduktan sonra, **'handleSubmit'** metodu çalıştırılarak dosya IPFS' e yüklenir ve elde edilen hash değeri blokzincire kaydedilir. Bu işlem sırasında, dosyanın kimlik bilgileri, IPFS hash değeri ve blokzincir işlem kimliği ilgili **'state'** değişkenlerinde saklanır. Kullanıcı, MetaMask ile giriş yaptığında ve dosya seçildiğinde dosya yükleme işlemi başlar. Yükleme işlemi tamamlandığında yüklenen dosyaların listesi ve bir önizlemesi, IPFS linkleri ve blokzincir işlem kimlikleri gösterilir. Bağlantı durumu, MetaMask hesabına bağlı olup olmama durumuna göre yeşil veya sarı bir simge ile gösterilir.



Fotoğraf 2.2.2.1. Uygulamanın giriş ekranı

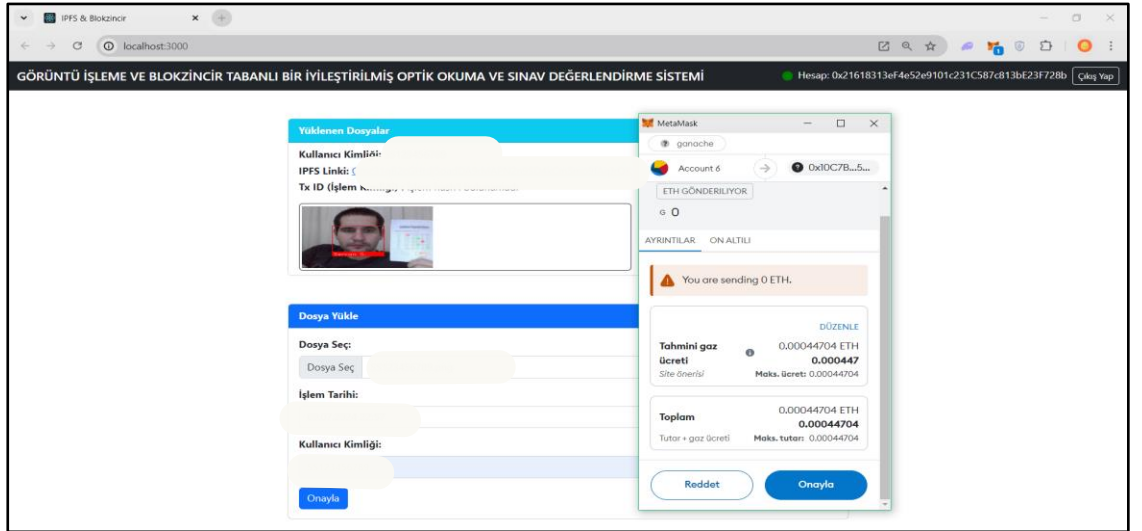
İlk aşamada, uygulama açıldığında MetaMask ile giriş yapılması gerektiği belirtilir. "MetaMask ile Giriş Yap" butonuna tıklandığında, **'handleMetaMaskLogin'** fonksiyonu çağrılır. Fonksiyon, Web3 sağlayıcısını (MetaMask) kullanarak kullanıcıdan hesaplarını bağlamasını ister. MetaMask hesabı bağlandıktan sonra, hesap bilgileri ve ağ bilgileri alınır ve uygulamaya ait **'state'** kaydedilir.



Fotoğraf 2.2.2.2. Ganache test hesabı ile Metamask bağlantısı

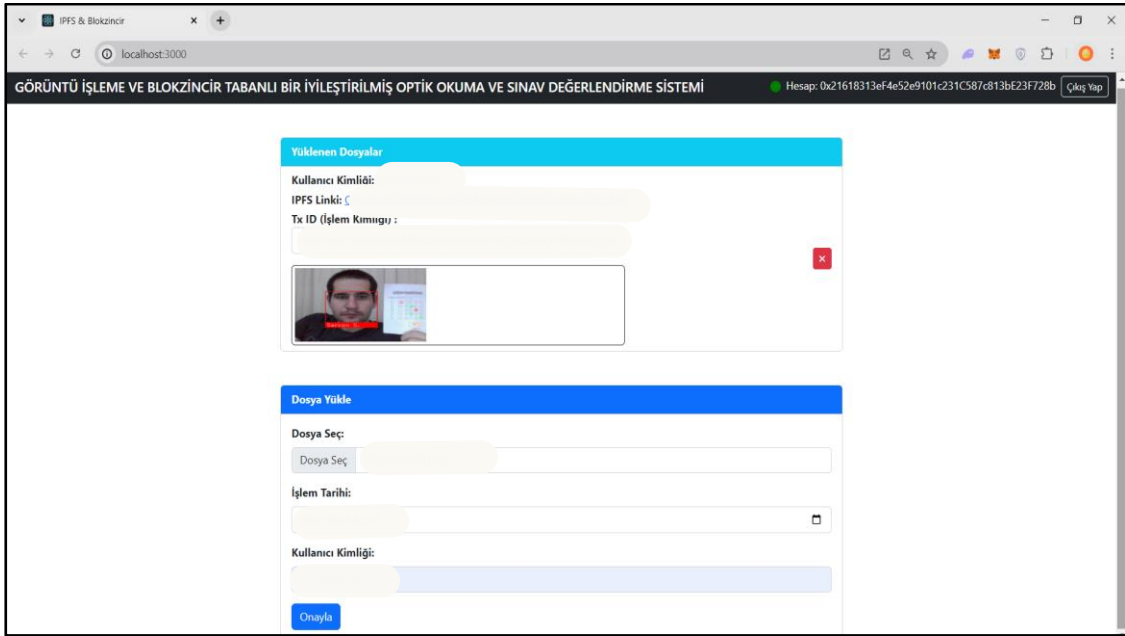
MetaMask hesabı bağlandıktan sonra, Web3 sağlayıcısı '**Web3.givenProvider**' kullanılarak ağ ve hesap bilgileri alınır. Bu bilgiler arasında hesabın adresi, bağlı olduğu ağ türü ve ağ ID'si bulunur. Bu bilgiler '**state**' olarak kaydedilir ve aynı zamanda akıllı sözleşme (smart contract) örneği oluşturulur ve state'e eklenir. Böylelikle MetaMask'ın başarıyla bağlandığı ve kullanıcının hesabının aktif olduğu doğrulanır.

Uygulama, test ortamında çalıştığından, yerel blokzincir ağı olan Ganache ile entegrasyon sağlanmıştır. Ganache, geliştirme ve test amaçlı kullanılan bir Ethereum ağıdır. Ganache kullanılarak, geliştiriciler kendi bilgisayarlarında özel bir Ethereum ağı oluşturabilir ve bu ağda test işlemleri yapabilirler. Ganache geliştiricilere sanal Ether (ETH) ile işlem yapma olanağı tanır. Böylece, kullanıcı MetaMask aracılığıyla Ganache ağına bağlandığında, test hesabı oluşturulmuş olur ve bu hesap üzerinden işlemler gerçekleştirilir. Fotoğraf 2.2.2.2.' de 100 ETH bulunan bir test hesabı kullanılmıştır.



Fotoğraf 2.2.2.3. Dosya yükleme ve onay ekranı

"Dosya Seç" butonuna tıklandığında, **'captureFile'** fonksiyonu çağrılır. Bu fonksiyon, seçilen dosyayı alır, bir **'FileReader'** nesnesi kullanarak dosyayı okur ve dosya verisini (buffer state'e) kaydeder. Aynı zamanda, bilgisayarın güncel tarih ve saat bilgisi **'setCurrentDateTime'** fonksiyonu ile alınır ve state'e kaydedilir. Kimlik bilgileri girildikten sonra "Onayla" butonuna basılır. Buton, **'handleSubmit'** fonksiyonunu tetikler ve IPFS' e dosya yükleme süreci başlar. Dosya IPFS' e yüklenir ve dosyanın hash' i alınır. Daha sonra bu hash, akıllı sözleşmeye kaydedilir ve MetaMask üzerinden işlem ücreti ve detayları gösterilir.

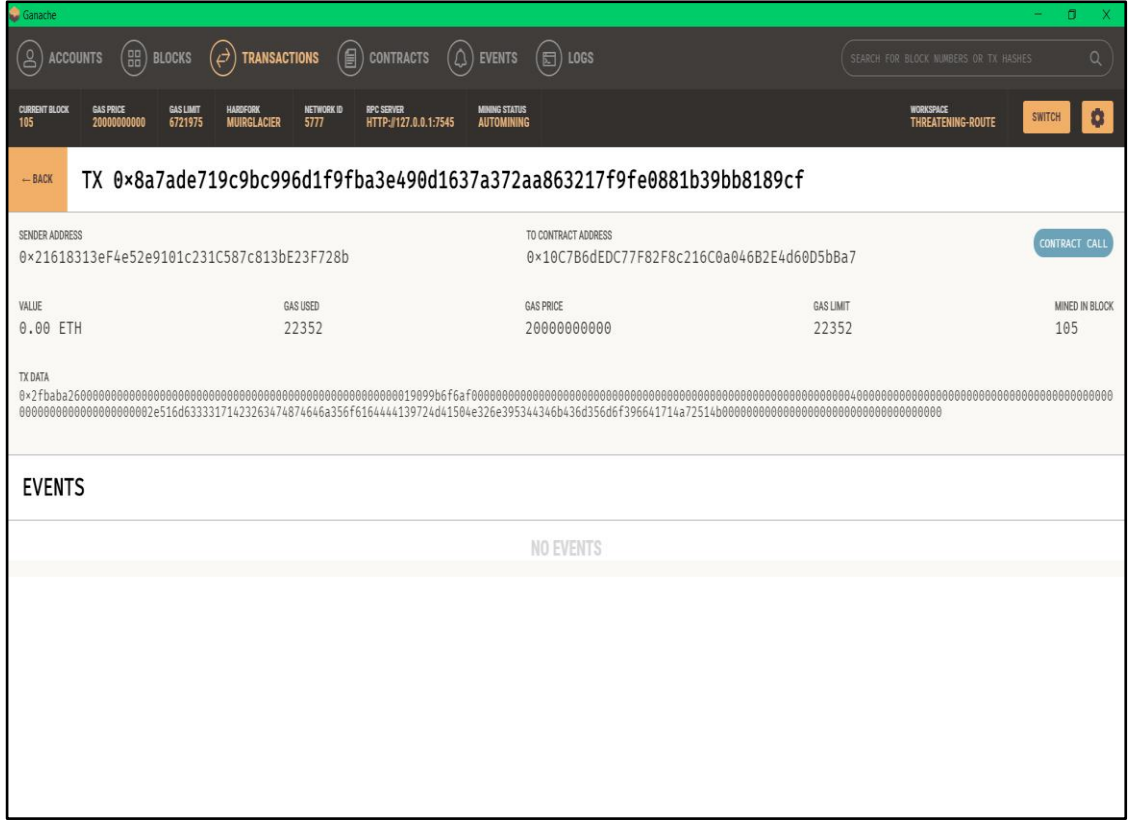


Fotoğraf 2.2.2.4. Yüklenen dosya bilgilerinin görüntülenmesi

Dosya başarıyla IPFS'e yüklendikten ve Ethereum blokzincirindeki işlem tamamlandıktan sonra, uygulama **'timeNhash'** adlı state'i günceller. Bu state, kullanıcıların yükledikleri dosyaların detaylarını tutar ve uygulamanın arayüzünde gösterilir.

IPFS Linki: IPFS üzerinde depolanan dosyanın benzersiz adresi (path). IPFS, dağıtık bir dosya sistemidir ve bu link, dosyanın nerede saklandığını ve nasıl erişilebileceğini gösterir. Kullanıcılar bu link aracılığıyla ilgili dosyaya erişebilir.

İşlem Kimliği (Transaction ID): Ethereum blokzinciri üzerinde yapılan işlemin kimliğidir (Tx ID). Her işlem için benzersiz olan bu kimlik, dosyanın IPFS linkinin hangi blokzincir işlemiyle ilişkilendirildiğini gösterir. Kullanıcılar, işlem kimliği aracılığıyla işlemin durumunu blokzincirde izleyebilir ve dosyanın güvenli bir şekilde saklandığını doğrulayabilir.



Fotoğraf 2.2.2.5. Ganache hesabı üzerinden yapılan işlemin görüntülenmesi

Son aşamada, Ganache üzerinden yapılan işlemler ve bu işlemlerin kimlikleri (transaction IDs) görüntülenir. Kullanıcı yapılan işlemlerin detaylarını MetaMask üzerinden veya doğrudan Ganache üzerinden takip edebilir. Blokzincirde görüntü verilerinin depolanması işlemi de tamamlandıktan sonra geriye gizlilik ve güvenliği sağlanmasındaki önemli aşamalardan olan verilerin şifrlenmesi işlemi kalmış oluyor.

2.2.3. Verilerin Şifrlenmesi

Kriptografi, bilgi güvenliği alanında önemli olan ve bilgilerin gizliliğini, bütünlüğünü ve doğruluğunu korumak için kullanılan matematiksel ve algoritmik yöntemleri kapsayan bir disiplindir. Temel amacı iletilen ve saklanan bilgilerin sadece yetkili kişiler tarafından erişilebilir olmasını sağlamak olan kriptografi, binlerce yıllık geçmişe sahip olup, antik zamanlardan günümüze kadar gelişim göstermiştir. Başlangıçta, genellikle askeri iletişimde kullanılan şifreleme yöntemleri ile başlayan kriptografi, modern bilgi teknolojileriyle birlikte dijital dünyada önemli bir yer edinmiştir. Günümüzde, internet üzerinden yapılan iletişimden finansal işlemlere kadar geniş alanlarda kullanılan kriptografi, günlük hayatımızın vazgeçilmez bir parçası haline gelmiştir.

Kriptografi temelde iki ana kategoriye ayrılır: simetrik ve asimetrik (veya genel anahtarlı) kriptografi. Simetrik kriptografi, aynı anahtarın hem şifreleme hem de şifre çözme işlemlerinde kullanıldığı yöntemdir. Hızlı işlem yapabilmesi ve veri şifreleme sırasında sağladığı yüksek güvenlik seviyesi ile bilinir. Asimetrik kriptografi ise, farklı iki anahtarın kullanıldığı bir sistemdir: bunlar genel anahtar ve özel anahtardır. Dijital imzalar, anahtar değişimi ve güvenli iletişim gibi uygulamalarda kullanılır. Asimetrik şifreleme alanındaki önde gelen algoritmalar arasında RSA (Rivest-Shamir-Adleman) ve ECDSA (Elliptic Curve Digital Signature Algorithm) bulunmaktadır. RSA, büyük asal sayıların çarpımı şeklinde ifade edilen sayılarla işlem yapar ve genellikle anahtar değişimi, dijital imza ve kimlik doğrulama gibi uygulamalarda tercih edilir.

Anahtar uzunlukları genellikle 1024-bit ile 4096-bit arasında değişebilir ve güvenliği kanıtlanmış bir algoritmadır. ECDSA ise eliptik eğri kriptografisi temelinde çalışır ve daha küçük anahtar uzunluklarıyla daha yüksek güvenlik sağlayabilir. Özellikle mobil cihazlar ve kaynak kısıtlı ortamlarda etkin bir şekilde kullanılır.

Simetrik şifreleme alanında öne çıkan iki önemli algoritma ise DES (Data Encryption Standard) ve AES (Advanced Encryption Standard) olarak bilinir. DES, 1970'lerde geliştirilmiş olan eski bir simetrik şifreleme algoritmasıdır. 56-bit uzunluğundaki anahtar kullanımıyla tanınır ve uzun süre güvenli bir seçenek olarak kabul edilmiştir. Ancak, günümüzde artan işlem gücü ve gelişmiş kırılma teknikleri nedeniyle güvenlik açıkları taşıdığı düşünülerek terk edilmiştir. DES'in yerini daha güçlü şifreleme algoritmaları almıştır. AES ise modern simetrik şifreleme algoritmalarının standartlarından biridir. AES'in 128-bit, 192-bit ve 256-bit olmak üzere farklı anahtar uzunlukları mevcuttur ve güçlü bir güvenlik sağlamasıyla bilinir. Hızlı işlem kapasitesi ve geniş uygulama yelpazesıyla modern bilgi teknolojilerinde yaygın olarak kullanılan bir şifreleme algoritmasıdır. AES hem veri şifreleme hem de güvenli iletişim gibi birçok uygulamada tercih edilen bir seçenektir. Bu avantajlarından dolayı tez çalışmasında AES şifreleme kullanılacaktır.

AES şifreleme ve şifre çözme işlemleri için '**PyCryptodome**' kütüphanesi kullanılacaktır. Ayrıca görüntü dosyalarını açma, işleme ve kaydetme işlemleri için araçlar sağlayan '**Pillow**' kütüphanesi de kullanılacaktır. Daha sonra blokzincir cüzdan adresi doğrulaması yapılarak görüntü verisinin şifresi çözülecek veya şifreli hali gösterilecektir.

```

from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import base64
from PIL import Image
import io

def pad(data):
    while len(data) % 16 != 0:
        data += b' '
    return data

def unpad(data):
    return data.rstrip(b' ')

```

Şekil 2.2.3.1. Görüntü verilerinin şifrlenmesi

İlk olarak gerekli kütüphaneler import edilerek başlanmıştır. Sonrasında görüntü verisinin boyutunu 16 baytlık bloklara tamamlayacak olan padding işlemi için pad ve unpad fonksiyonları tanımlanmıştır.

```

def encrypt_image(image_data, key):
    cipher = AES.new(key, AES.MODE_ECB)
    padded_data = pad(image_data)
    encrypted_data = cipher.encrypt(padded_data)
    return base64.b64encode(encrypted_data).decode('utf-8')

```

Şekil 2.2.3.2. Görüntü verilerinin şifrlenmesi-2

Görüntü verisi AES algoritması ile şifrenilmiş, base64 formatında kodlanmış bir string olarak döndürülmüştür. Bu, şifrenilmiş verinin dosyada saklanması veya başka sistemlere iletilmesini kolaylaştırır.

```

def decrypt_image(encrypted_data, key, wallet_address,
authorized_wallet_address):
    if wallet_address != authorized_wallet_address:
        raise ValueError("Yetkisiz cüzdan adresi!")

    cipher = AES.new(key, AES.MODE_ECB)
    encrypted_data = base64.b64decode(encrypted_data)
    decrypted_data = cipher.decrypt(encrypted_data)
    return unpad(decrypted_data)

```

Şekil 2.2.3.3. Görüntü verilerinin şifrlenmesi-3

Şifrelenen verinin şifresini çözmek için ‘decrypt_image’ fonksiyonu tanımlanmıştır. Aynı zamanda blokzincir cüzdan adresi doğrulaması yapılarak yetkisiz

erişimler engellenmektedir. Fonksiyon ilk olarak, verilen cüzdan adresinin yetkili cüzdan adresi olup olmadığını doğrular. Eğer cüzdan adresi yetkili değilse, "Yetkisiz cüzdan adresi!" hatası görüntülenir işlem durdurulur. Bu adımdan sonra, AES şifreleme objesi oluşturulur ve base64 formatında kodlanmış şifrelenmiş veri çözülerek orijinal şifrelenmiş byte verisine dönüştürülür. Ardından, AES şifreleme objesi kullanılarak şifrelenmiş veri çözülür ve şifresiz byte verisi elde edilir. Son olarak, şifresi çözülmüş veri, orijinal blok boyutuna getirilir ve veriye eklenmiş olan ekstra baytlar kaldırılır.

```
def read_image_as_bytes(image_path):  
    with Image.open(image_path) as img:  
        img_byte_arr = io.BytesIO()  
        img.save(img_byte_arr, format=img.format)  
        return img_byte_arr.getvalue()  
  
def save_bytes_as_image(image_data, output_path):  
    img = Image.open(io.BytesIO(image_data))  
    img.save(output_path)
```

Şekil 2.2.3.4. Görüntü verilerinin şifrenmesi-4

‘read_image_as_bytes’ fonksiyonu, belirtilen dosya yolundaki görüntü dosyasını okuyarak byte formatına dönüştürür. Görüntü verisi açıldıktan sonra, ‘io.BytesIO()’ kullanılarak bir byte dizisi akışı oluşturulur. Ardından, ‘save_bytes_as_image’ fonksiyonu, byte formatındaki görüntü verisini belirtilen dosya yoluna kaydeder.

```
key = get_random_bytes(16) # 128-bit anahtar  
authorized_wallet_address =  
"0x21618313eF4e52e9101c231C587c813bE23F728b"  
  
image_path = 'örnek_görüntü.png'  
output_encrypted_path = 'şifrelenmiş_görüntü.txt'  
output_decrypted_path = 'çözülmüş_görüntü.png'  
  
image_data = read_image_as_bytes(image_path)  
encrypted_data = encrypt_image(image_data, key)  
print("Şifrelenmiş Veri:", encrypted_data[:100])  
with open(output_encrypted_path, 'w') as file:  
    file.write(encrypted_data)
```

Şekil 2.2.3.5. Görüntü verilerinin şifrenmesi-5

İlk olarak, **'get_random_bytes(16)'** fonksiyonu ile 128-bit uzunluğunda bir rastgele anahtar (key) oluşturulur. Yetkili cüzdan adresi olarak daha önce Ganache test ağından sağlanan cüzdan olan **'0x21618313eF4e52e9101c231C587c813bE23F728b'** atanır. **'read_image_as_bytes(image_path)'** fonksiyonu, belirtilen dosya yolundaki görüntü dosyasını okuyarak byte formatına dönüştürür. Byte verisi şifrelenir ve şifrelenmiş veri elde edilir. Şifrelenmiş verinin ilk 100 karakteri konsolda görüntülenir. **'with'** anahtar kelimesi, dosya işlemlerinde otomatik dosya kapatma işlemini yönetir ve dosya işlemi tamamlandığında dosyanın kapatılmasını sağlar.

```
wallet_address = "0x21618313eF4e52e9101c231C587c813bE23F728b"
try:
    decrypted_data = decrypt_image(encrypted_data, key,
wallet_address, authorized_wallet_address)
    save_bytes_as_image(decrypted_data, output_decrypted_path)
    print("Şifre çözülmüş ve görüntü kaydedilmiştir:")
except ValueError as e:
    print(e)
```

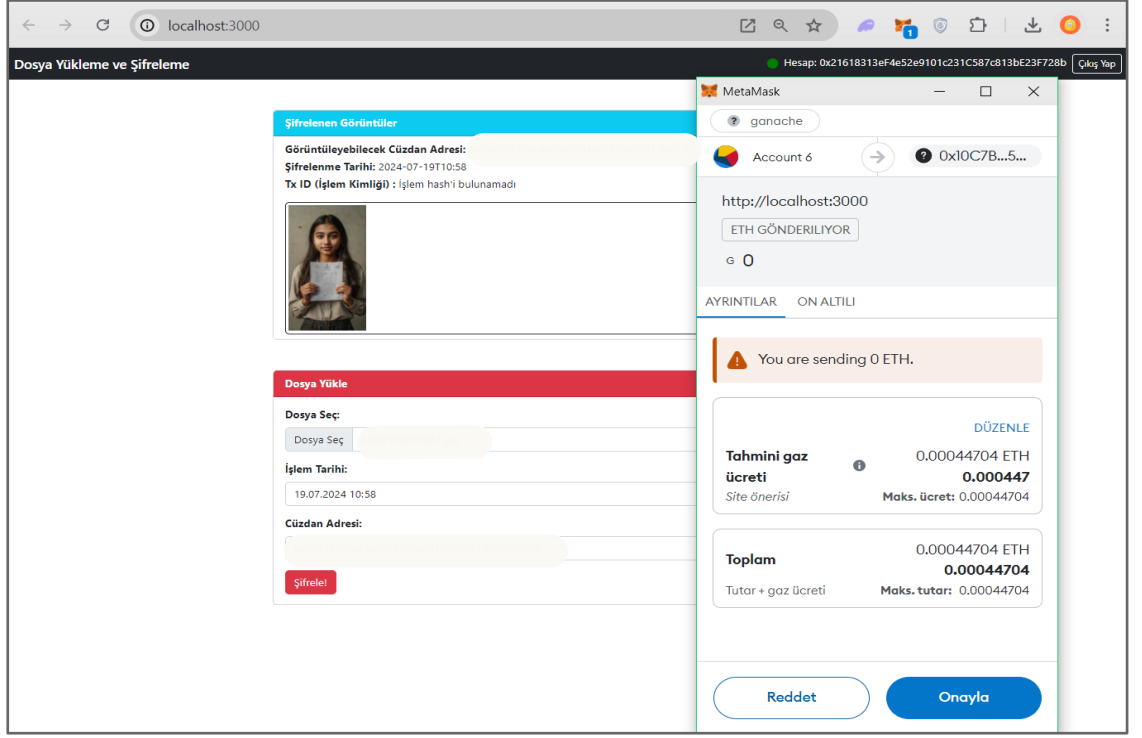
Şekil 2.2.3.6. Görüntü verilerinin şifrelenmesi-6

Burada, yetkili cüzdan adresi ayarlandıktan sonra **'decrypt_image'** fonksiyonu çağrılarak; şifrelenmiş veri, şifreleme anahtarı ve yetkili cüzdan adresi kullanılarak çözülür. Şifre çözme başarılı olursa, elde edilen veriler belirtilen dosya yoluna kaydedilir ve **'Şifre çözülmüş ve görüntü kaydedilmiştir.'** mesajı yazdırılır. Eğer bir hata var ise, **'ValueError'** ile hata mesajı yazdırılır.

```
wallet_address = "0xUnauthorizedWalletAddress"
try:
    decrypted_data = decrypt_image(encrypted_data, key,
wallet_address, authorized_wallet_address)
    save_bytes_as_image(decrypted_data, output_decrypted_path)
    print("Cüzdan adresi tanımlı olmadığından şifreli halini
görüntülemektesiniz:", output_decrypted_path)
except ValueError as e:
    print(e)
```

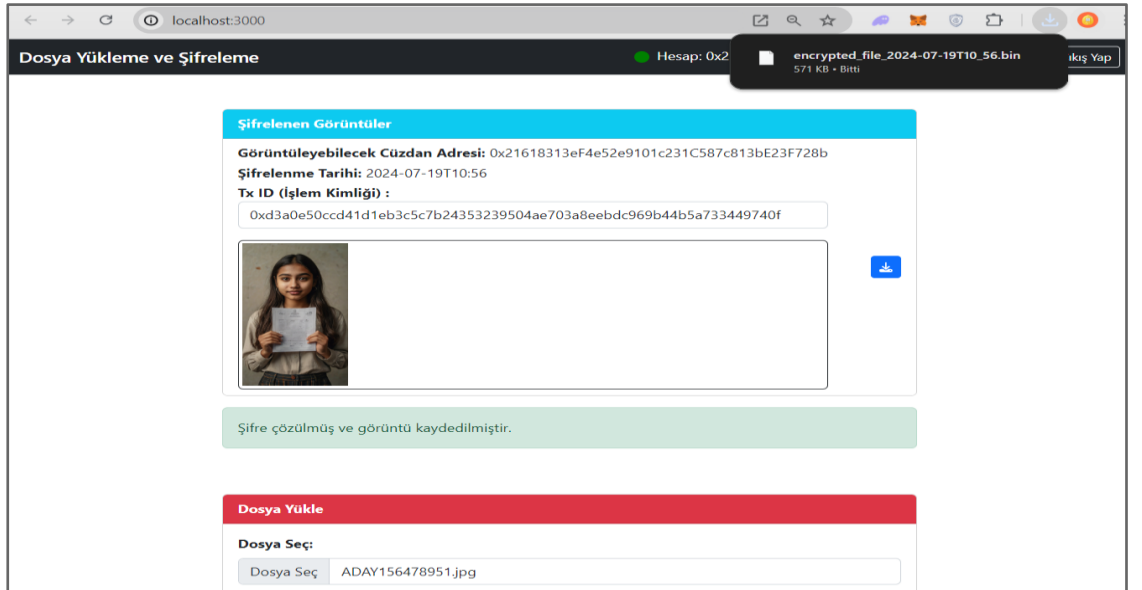
Şekil 2.2.3.7. Görüntü verilerinin şifrelenmesi-7

Son kısımda ise, **'wallet_address'** değişkeni, rastgele bir cüzdan adresi olan **"0xUnauthorizedWalletAddress"** ile tanımlanır. Şifre çözme işlemi tekrar denenir ve bu durumda **'ValueError'** durumu yakalanır, çünkü cüzdan adresi yetkisizdir. **'Cüzdan adresi tanımlı olmadığından şifreli halini görüntülemektesiniz.'** mesajı yazdırılır.



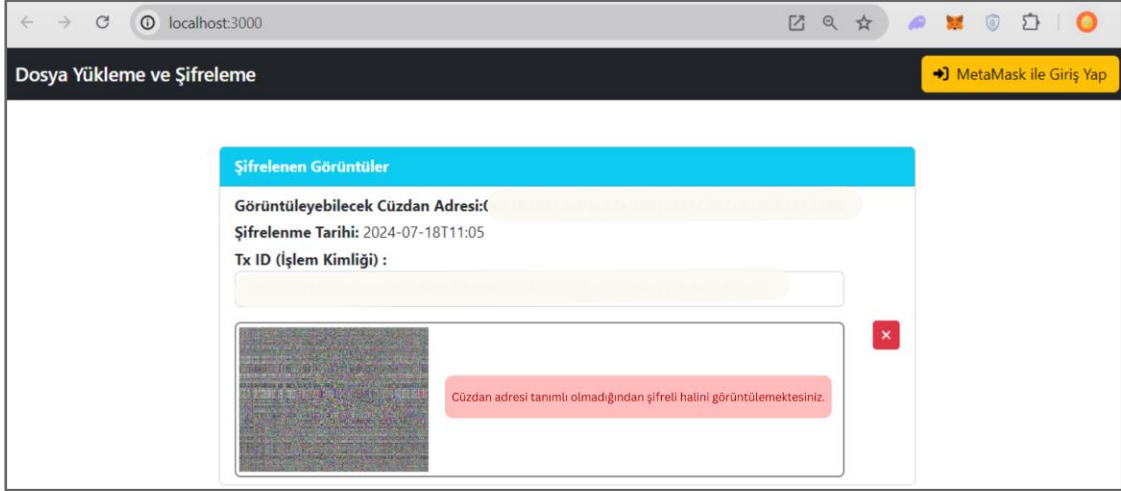
Fotoğraf 2.2.3.1. Görüntü verisini şifreleme işleminin görüntülenmesi

Şifrelenmek istenen görüntü verisi ‘Dosya Seç’ seçeneği kullanılarak seçilir ve sistem tarafından otomatik zaman damgası işlem tarihi olarak işlenir. Cüzdan adresi, şifrelenmiş görüntünün şifresini çözmek için kullanılacak olan adrestir. Örnek uygulama için Ganache test ağından sağlanan cüzdan adresi kullanılmıştır. Şifrele butonuna tıklandığında işlem gerçekleşir ve görüntü şifrelenir.



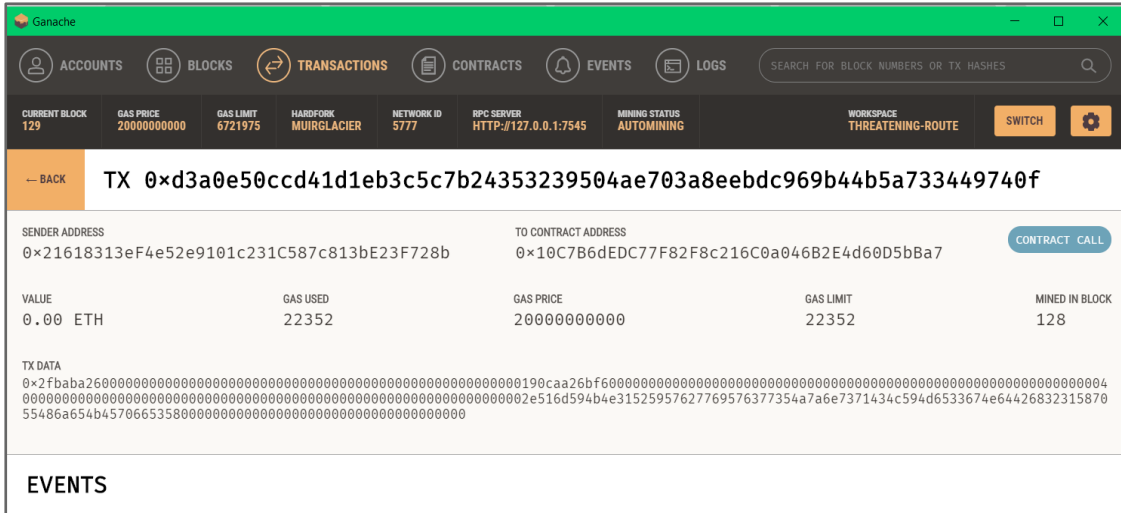
Fotoğraf 2.2.3.2. Görüntü verisinin şifre çözme işleminin görüntülenmesi

Görüntünün şifresini çözebilmek için doğru cüzdan adresi ile giriş yapılmış olmalıdır. Örnekte görülen **0x21618313eF4e52e9101c231C587c813bE23F728b** cüzdan adresi ile giriş yapıldığında görüntü şifresi çözülerek ve kaydedilmiştir. Yine başarılı işlemin Tx ID' si şifreleme tarihi altında görüntülenmektedir.



Fotoğraf 2.2.3.3. Başarısız olan bir şifre çözme işlemi

Metamask girişi yapılmadığında veya tanımlı olmayan cüzdan adresi ile giriş yapıldığında, görüntü verisi şifreli olarak görüntülenmektedir. Yapılan her işlem (başarısız da olsa) blokzincirde kayıt altına alınmaktadır.



Fotoğraf 2.2.3.4. Ganache hesabı üzerinden şifreleme işleminin görüntülenmesi

Son aşamada ise Ganache üzerinden, yapılan işlemler ve bu işlemlerin kimlikleri görüntülenebilir. **Fotoğraf 2.2.3.2.** te gerçekleştirilen başarılı şifre çözme işleminin Tx ID' si Ganache ortamında doğrulanmıştır. Görüntü işleme modülleri ve blokzincir modülleri başarıyla tamamlandıktan sonra akıllı sınav güvenliği sisteminin analizi ve sistem uygulama aşamaları üçüncü bölümde incelenecektir.

ÜÇÜNCÜ BÖLÜM

AKILLI SINAV GÜVENLİĞİ

3.1. Akıllı Sınav Güvenliği Alanında Yapılan Çalışmaların Analizi

Çevrimiçi sınav sistemlerinde birçok güvenlik açığının bulunduğu ve yanlış şifre kullanarak kopya çekmenin mümkün olduğunu belirten bir çalışmada, daha güvenli bir sistem önerilmektedir [30].

Çevrimiçi sınavları yürüten kurumlar, merkezi sunuculardan bir kopya almaya gerek kalmadan verileri kolayca düzenleyebilirler. Önerilen blokzincir çerçevesi, veri güvenliğini artırmakta ve kullanıcılar veya üçüncü taraflar arasında kopya çekmeyi ortadan kaldırmaktadır [31].

Çevrimiçi sınavlarda kimlik sahtekarlığını önlemek için geliştirilen bir başka sistem önerisi, sınavı yasa dışı adayların almasını engellemeyi amaçlamaktadır. Ayrıca, sınav sonunda adaylara e-sertifika verilmesi hedeflenmiştir. Önerilen sistemde, sınav süreçlerinde güvenliği artırmak ve kimlik doğrulama süreçlerini iyileştirmek için önemli bir adım teşkil etmektedir [32].

Blokzincir ve Merkle ağaç yapısını kullanan bir çalışma, kullanıcı profillerinin güvenli bir şekilde saklanması ve ‘Profil Tabanlı Kimlik Doğrulaması’ çerçevesini geliştirmek için etkili bir kimlik doğrulama süreci önermiştir. Çalışma, e-öğrenme platformlarında güvenliği ve veri bütünlüğünü artırmayı amaçlamaktadır [33].

Blokzincir teknolojisinin çevrimiçi öğrenmeyi nasıl dönüştürdüğünü ve sürdürülebilirlikle ilişkili üç ana tema olan standart bilgi, davranış modeli ve dijital rozet ile nasıl bağlantılı olduğunu inceleyen bir çalışmada, önemli bulgular sunulmuştur [34].

Sonuç olarak, sınav sistemlerinin güvenliği, blokzincir teknolojisi ile önemli ölçüde artırılabilir. Bu çalışmalar, blokzincirin veri güvenliği, kimlik doğrulama ve kullanıcı gizliliği gibi kritik alanlarda sağladığı katkıları gözler önüne sermektedir. Blokzincir teknolojisinin eğitim sektöründeki potansiyeli, daha güvenli ve verimli sınav sistemlerinin geliştirilmesi için umut verici bir yol haritası sunmaktadır. Fakat çalışmalar incelendiğinde sistemlerin genellikle çevrimiçi sınavlar için uygulandığı tespit edilmiştir.

Bu tez çalışmasında benzer uygulamalar olmakla birlikte, görüntü işleme ve blokzincir tabanlı yüz yüze veya çevrimiçi sınavlara rahatlıkla uyarlanabilecek bir sınav güvenliği sistem önerisi geliştirilecektir.

3.2. Akıllı Sınav Güvenliği Sisteminin Uygulama Aşamaları

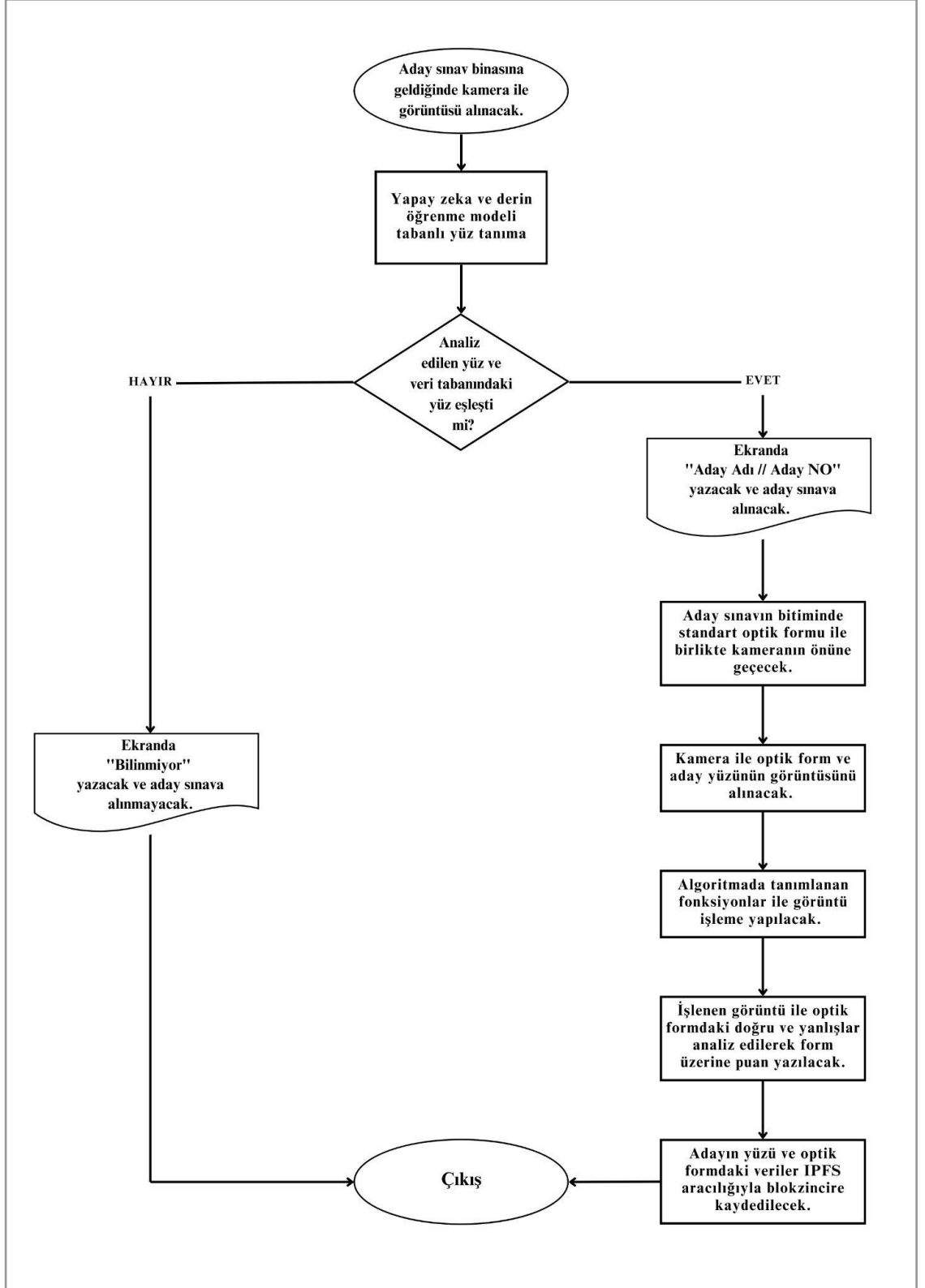
Günümüzde akıllı sınav güvenliği, hem adayların sınav süreçlerine olan güvenini artırmak hem de sınav sonuçlarının doğruluğunu ve bütünlüğünü sağlamak açısından büyük önem taşımaktadır. Önerilen sistem, yapay zekâ ve blokzincir teknolojilerini entegre ederek yenilikçi ve güvenli bir sınav yönetim sistemi sunmaktadır. Yüz tanıma teknolojileri ve görüntü işleme algoritmaları sayesinde, adayların kimlik doğrulama süreçleri hassas bir şekilde yürütülmekte ve sınav sonuçlarının manipülasyona karşı korunması sağlanmaktadır.

Özellikle sınav sonunda görüntü işleme aracılığıyla optik form üzerinden yapılan anlık değerlendirmeler ve bu verilerin blokzincir teknolojisi ile güvence altına alınması, sistemin şeffaflık ve güvenilirlik açısından üst düzey performans sergilemesini mümkün kılmaktadır. Bu gelişmiş güvenlik sistemi, dijital cüzdan adresleri aracılığıyla panel erişimi ve adayların sadece kendi verilerine ulaşabilmesi gibi ek güvenlik katmanları ile desteklenmektedir.

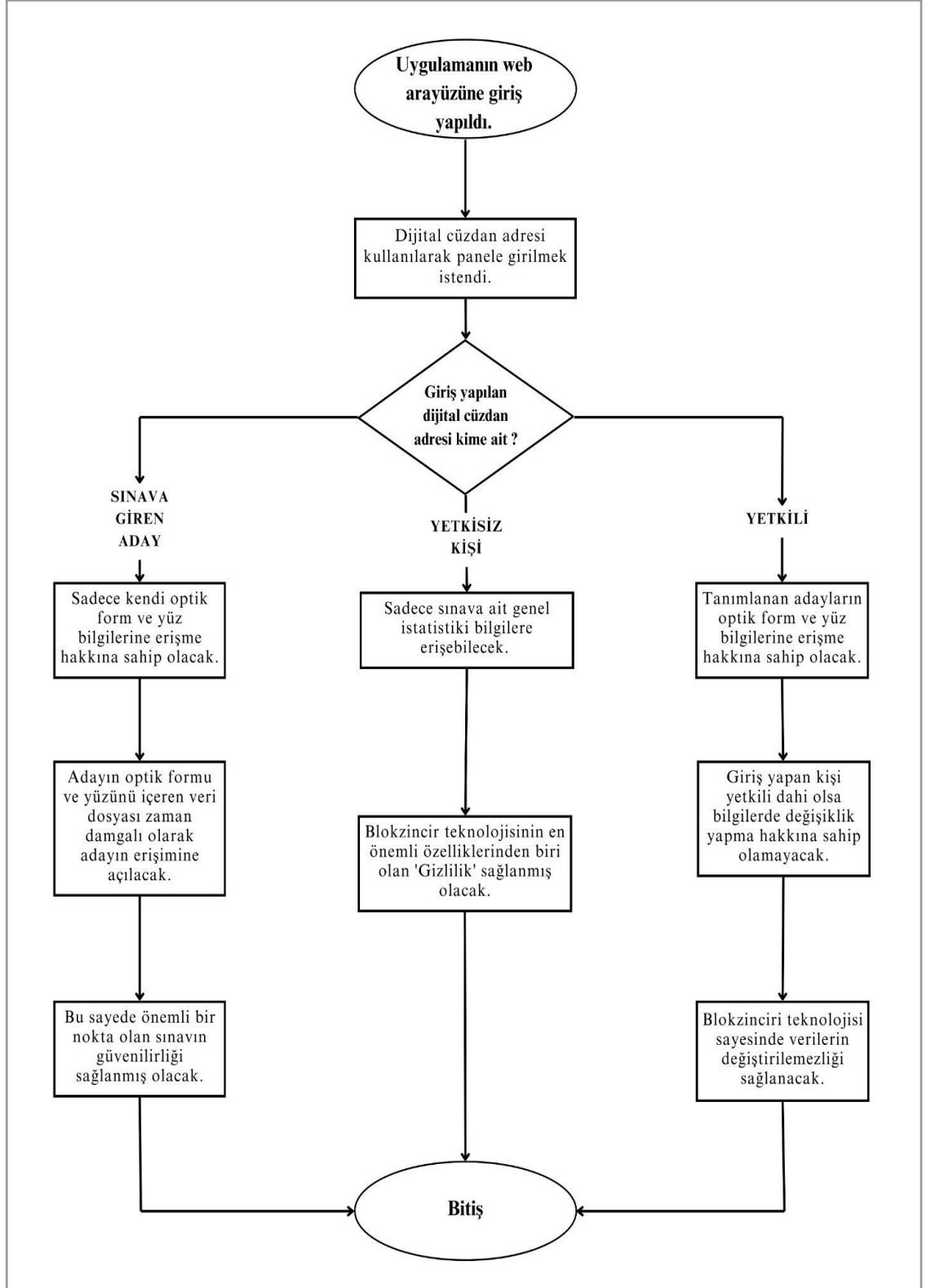
Aday sınav binasına geldiğinde, ilk olarak yüz tanıma algoritması ile kimlik doğrulaması yapılır. Adayın yüzü ve veritabanındaki kayıtlı yüz analiz edilerek eşleşirse, ekranda "**Aday Adı // Aday NO**" bilgisi görüntülenir ve aday sınava alınır. Eşleşme sağlanmazsa aday tanınmadığı için sınava alınmamış olur. Sınavın bitiminde, aday standart optik formunu doldurarak kamera önüne geçer. Kamera hem adayın yüzünü hem de optik formu tek seferde kaydeder. Görüntüler, tanımlanan algoritmalar aracılığıyla işlenir ve optik formdaki doğru ve yanlışlar analiz edilerek form üzerine puan yazılır. İşlenen bu veriler, zaman damgalı olarak IPFS aracılığıyla blokzincire kaydedilir, böylece verilerin değiştirilemezliği ve güvenilirliği sağlanmış olur.

Blokzincir teknolojisi, sınav güvenliğini sağlamada kritik bir rol oynar. Adaylar sadece kendi optik form ve yüz bilgilerine erişim hakkına sahip olurken, yetkililer adayların optik form ve yüz bilgilerine erişebilir ancak bu bilgilerde değişiklik yapamazlar. Bu sayede, sınav sonuçlarının manipüle edilmesi önlenir ve sınavın güvenilirliği artırılır.

Aşağıda, bu akıllı sınav güvenliği sisteminin uygulama aşamalarını hem sınav anı hem de sınav sonrası olarak uygulama aşamalarını gösteren akış diyagramları sunulmuştur.



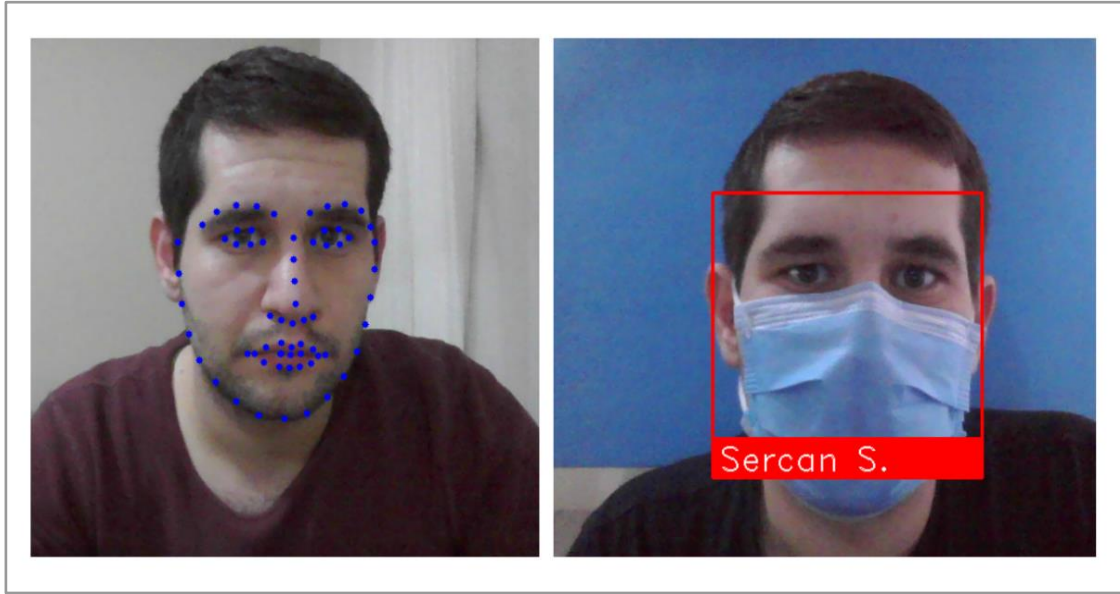
Şekil 3.2.1. Sınav anı uygulamaları akış diyagramı



Şekil 3.2.2. Sınav sonrası uygulamaları akış diyagramı

Fotoğraf 3.2.1. 'de görüldüğü üzere aday sınav binasına geldiğinde kamera ile görüntüsü alınarak, algoritma yardımıyla derin öğrenme modelini kullanarak yüzdeki

noktaları çıkarıyor. Tespit edilen yüzü veritabanına eklenen yüz ile karşılaştırarak yorumluyor. Ayrıca adayın maske takması yüz tanımayı engellemiyor.



Fotoğraf 3.2.1. 68 noktalı yüz tanıma modeli ve maskeli yüz tanıma

Yüz tanıma modeli birbirine çok benzeyen ikizlerin ayırt edilmesinde dahi başarılı sonuçlar ortaya çıkarmaktadır. Böylece aday yerine ikizinin bile sınava girmesi imkânsız hale gelmiş olacaktır.



Fotoğraf 3.2.2. Yüz tanıma modeli ile ikizlerin ayırt edilmesi

Adaylar sınav binasına yüz tanınması gerçekleştirilip alındıktan sonra artık sınavın uygulanması aşamasına geçilecektir. Bu noktada sınava özel hazırlanmış olan optik formlar adaylara verilecek ve işaretlemelerini bu forma yapmaları istenecektir. Algoritmada standart optik formda her sorunun doğru cevabı tanımlanmış olacaktır.

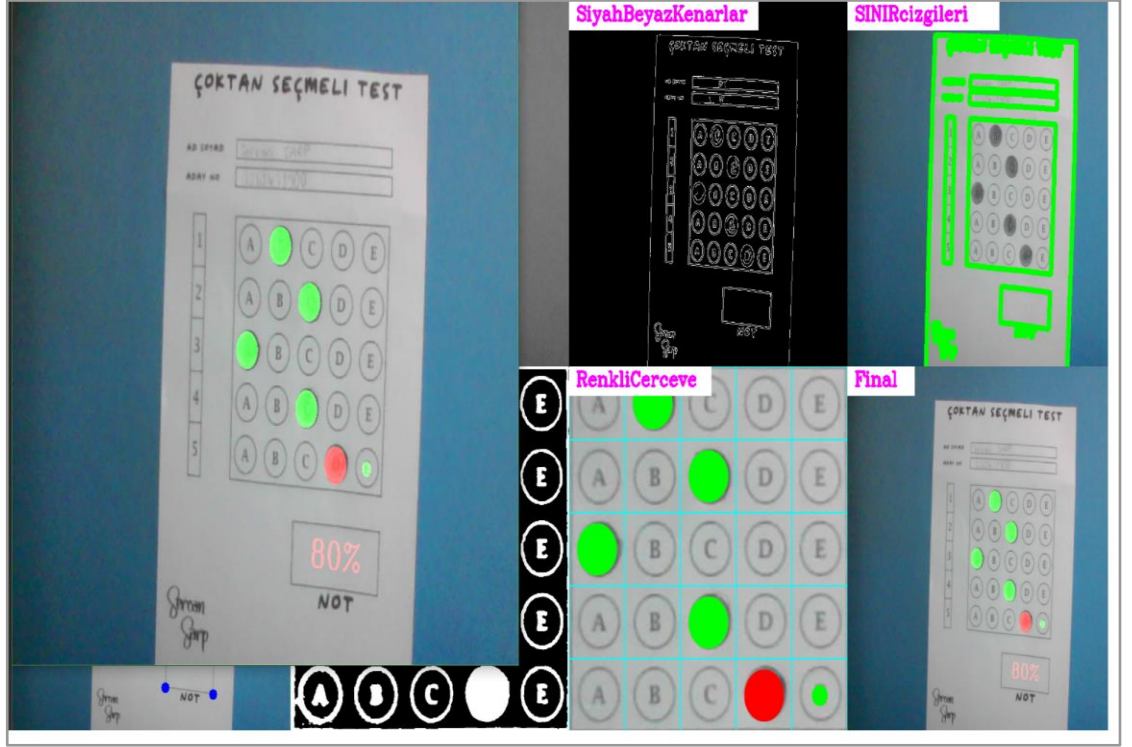
Aşağıda 5 soruluk ve her sorusunda 5 seçenek bulunan bir sınava göre hazırlanan örnek standart optik formun bir görseli paylaşılmıştır.

ÇOKTAN SEÇMELİ TEST					
AD SOYAD					
ADAY NO					
1	A	B	C	D	E
2	A	B	C	D	E
3	A	B	C	D	E
4	A	B	C	D	E
5	A	B	C	D	E
NOT					

Sercan Sarp

Fotoğraf 3.2.3. Standart optik formun görüntüsü

1.2.2. Optik Okuma ve Değerlendirme Modülünde paylaşılan algoritma ile optik okuma gerçekleştirilerek Fotoğraf 3.2.4’ deki sonuçlara ulaşılmıştır. Optik forma göre soruların doğru cevapları ‘1=A, 2=C, 3=A, 4=C ve 5=E’ olarak tanımlanmıştır. Sınava giren aday 1, 2, 3 ve 4. sorulara doğru yanıt vermesine rağmen, 5. soruyu yanlış cevapladığı için sınavda yüzde 80 oranında başarıya ulaşmıştır. Arka planda ise tanımlanan fonksiyonların çalışma prensipleri görülmektedir.



Fotoğraf 3.2.4. Standart optik form üzerinde yapılan görüntü işleme aşamaları

Görüntü işlemler hem girişteki yüz tanıma hem de optik okuma ve değerlendirme tarafında tamamlandıktan sonra geriye adayın optik formu ile kameranın karşısına geçmesi kalıyor. Aday optik formu ile kamera karşısına geçince hem adayın yüzü hem de optik formunun puanlanmış görüntüsü ekrana yansiyacak, aynı zamanda bu görüntü kaydedilecek. Kaydedilen görüntü blokzincir sayesinde saklanacak.



Fotoğraf 3.2.5. Aday yüz tanınması ve optik formunun değerlendirilmesi

Şekil 3.2.2.' deki sınav sonrası akış diyagramında bahsedildiği üzere sınav sonrası ekranlar için üç farklı yetki tipi tanımlanmıştır. İlk yetki tipi, cüzdanına herhangi bir yetki veya aday tipi tanımlanmamış olan kullanıcıları kapsamaktadır. Bu kullanıcılar sisteme giriş yaptıklarında, herhangi bir adayın sınav sonucunu görüntüleyemezler. Bu tür cüzdanlar, sınav sonuçlarına erişim konusunda tamamen yetkisizdir.

Sonuç Açıklama Sistemi

Giriş yaptığınız cüzdan adresi bu sınava giren adaylar listesinde olmadığından sadece sınava ait istatistikleri görüntülemektesiniz !

ÖSYM

<https://www.osym.gov.tr/TR,24039/2022-kpss-lisans-genel-yetenek-genel-kultur-egitim-bilimleri-olan-bilgisi-ve-oabt-sinav-sonuclarina-iliskin-sayisal-bilgiler.html>

Tablo 1. 2022 KPSS Lisans (Genel Yetenek - Genel Kültür, Eğitim Bilimleri, Alan Bilgisi, ÖABT ve ÖABT-İmam-Hatip Lisesi Meslek Dersleri Öğretmenliği) Oturumlara Göre Aday Sayıları

	Pazar Sabah Oturumu (Genel Yetenek - Genel Kültür)	Pazar Öğleden Sonra Oturumu (Eğitim Bilimleri)	Cumartesi Sabah Alan Oturumu (Kamu Yönetimi, Uluslararası İlişkiler, Çalışma Ekonomisi ve Endüstri İlişkileri)	Cumartesi Öğleden Sonra Alan Oturumu (Hukuk, İktisat, Maliye)	Pazar Sabah Alan Oturumu (İşletme, Muhasebe, İstatistik)	ÖABT (1.Oturum)	ÖABT İmam-Hatip Lisesi Meslek Dersleri Öğretmenliği
Başvuran Aday Sayısı	1.316.495	476.974	37.301	56.270	42.396	383.781	50.282
Sınava Giren Aday Sayısı	1.180.552	421.140	27.886	42.349	31.484	341.917	45.933
Sınavı Geçerli Aday Sayısı	1.180.485	421.132	27.886	42.344	31.482	341.910	45.932
Sınavı Geçersiz Aday Sayısı	67	8	0	5	2	7	1
Sınava Girmeyen Aday Sayısı	135.943	55.834	9.415	13.921	10.912	41.864	4.349

Farklı bir sonuç görüntülemek için aşağıdaki butona tıklayınız.

Sonuç Açıklama Sistemi

Fotoğraf 3.2.6. Giriş yapan herhangi bir kişinin görüntüleyeceği ekran

*<https://www.osym.gov.tr/TR,24039/2022-kpss-lisans-genel-yetenek-genel-kultur-egitim-bilimleri-olan-bilgisi-ve-oabt-sinav-sonuclarina-iliskin-sayisal-bilgiler.html>

İkinci yetki tipi, cüzdanına sınav yetkilisi rolleri tanımlanmış olan kullanıcılar için geçerlidir. Yetkili kullanıcılar sisteme giriş yaptıklarında, sınava giren adayların sonuçlarını görüntüleyebilirler. Ayrıca, adayların yüz tanıma ve optik form doğrulama verilerine erişim sağlayabilirler. Ancak bu yetki tipi, kullanıcıların herhangi bir değişiklik veya düzeltme yapmalarına izin vermez. Sınav yetkilisi kullanıcılar, sınava ait tüm verileri görme yetkisine sahip olsalar da veriler üzerinde herhangi bir müdahalede bulunamazlar.



Fotoğraf 3.2.7. Sınav yetkililerinin görüntüleyeceği ekran

Üçüncü yetki tipi ise, cüzdanına kendi sınav sonucu ve yüz tanıma bilgisi tanımlanmış olan kullanıcıları kapsar. Kullanıcılar, sisteme giriş yaptıklarında yalnızca kendi sınav sonuçlarını görüntüleyebilir ve kendi yüz tanıma bilgilerine görüntüleyebilirler. Diğer adayların bilgilerine erişim sağlayamazlar ve herhangi bir değişiklik ve düzeltme yapma yetkileri bulunmaz.



Fotoğraf 3.2.8. Sınava giren adayın görüntüleyeceği ekran

Önerilen bu akıllı sınav güvenliği sistemi, sınav sonuçlarının güvenliğini sağlarken, aynı zamanda sınav sürecinin her aşamasında adayların haklarını korumaya yönelik bir yaklaşım sunar. Blokzincir teknolojisi sayesinde verilerin değiştirilmesi engellenir ve her türlü müdahale girişimi kayıt altına alınır.

SONUÇ

Bu tez çalışmasında görüntü işleme ve blokzincir teknolojilerini kullanarak sınav güvenliğini ve güvenilirliğini artıran bir sistem önerisi geliştirilmiştir. Önerilen bu sistem sayesinde yüz tanıma yapılarak sınava kaçak girişlerin önüne geçilecek, gerçek zamanlı optik okuma ve değerlendirme yapılarak adayların zamandan tasarruf etmesi sağlanacak ve sağlanan tüm bu verileri blokzincirde şifrelenerek kaydedilecektir. Çalışmada blokzincir ve görüntü işleme modülleri ayrı ayrı değerlendirilmiş ve birleşimlerinden verimli bir akıllı sınav güvenliği sistemi oluşturulmuştur.

Literatür ve mevcut sınav uygulamaları incelendiğinde görüntü işleme teknolojilerinin sadece yüz tanıma boyutunda ve nadiren sınavlarda kullanıldığı gözlemlenmiştir. Geliştirilen sistem önerisi ile yüz tanıma sisteminin bütün sınavlara uygulanması mümkün olabilecektir.

Bir diğer modül olan optik okuma ve değerlendirme modülü, öğretim modellerinden programlı öğretimin anında dönüt ilkesi temel alınarak geliştirilmiştir. Bu modül sayesinde sınava giren adayın sınav bitiminde başarı yüzdesini görmesi ve zamandan tasarruf etmesi sağlanabilecektir.

Blokzincir cüzdan adresleri geçmiş uygulamalara göre daha efektif kullanılmış olup; ekstra bir güvenlik katmanı oluşturulmuştur. Sadece doğru cüzdan adresi bağlıyken sistemin kullanılabilir olması ile gizlilik ve güvenlik bir üst seviyeye taşınmıştır.

Özellikle görüntü kriptolojisi konusunda literatüre yeni kavramlar kazandırılabilir. Görüntü verilerinin şifreli haldeyken dahi işlenebildiği homomorfik şifreleme algoritmaları geliştirilebilir.

Depolama konusunda yetersiz kalınması durumunda merkezi ve dağıtık depolamanın bir arada kullanılabildiği hibrit depolama çözümleri kullanılabilir. Örneğin büyük veri dosyaları ve dokümanlar merkezi depolamada saklanırken; meta veriler, işlem verileri ve akıllı sözleşmelerin sonuçları blokzincirde tutulabilir. Böylelikle sistem tam merkeziz veya tek merkezli olmaktan kurtarılabilir.

KAYNAKLAR

- [1] I. Kırbaş, "Blokzinciri teknolojisi ve yakın gelecekteki uygulama alanları," *Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, cilt.9, sayı. 1, ss. 75-82, 2018, doi:10.29048/makufebed.365066.
- [2] B. Sezer, S. Topal, U. Nuriyev, "Tpp Supply: A traceable and privacy-preserving blockchain system architecture for the supply chain," *Journal of Information Security and Applications*, vol. 66, 2022, doi: 10.1016/j.jisa.2022.103116.
- [3] N. Diri, B. Yalçınkaya, "Blokzincir uygulamalarında kişisel veri problemi: Depolama riskleri ve Öneriler," *Bilgi Yönetimi*, cilt. 5, sayı. 1, ss. 47-67, 2022.
- [4] S. Sayeed, H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Applied Sciences*, vol. 9, no. 9, p. 1788, Sep. 2019, doi:10.3390/app9091788.
- [5] R. Rothblum, "Homomorphic encryption: From private-key to public-key," *Theory of Cryptography*, Y. Ishai, Ed. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 219-234, 2011.
- [6] M. Li, Y. Qin, B. Liu, X. Chu, "Enhancing the efficiency and scalability of blockchain through probabilistic verification and clustering," *Information Processing Management*, vol. 58, no. 5, 2021, doi: 10.1016/j.ipm.2021.102650.
- [7] G. Samtaş, M. Gülesin, "Sayısal görüntü İşleme ve farklı alanlardaki uygulamaları," *Ejovoc (Electronic Journal of Vocational Colleges)*, vol. 2, no. 1, pp. 85-91, 2012.
- [8] I. İnce, I. Senel, F. Yılmaz, "Image processing and analysis in health: Advantages, challenges, threats and examples," *Archives of Health Science and Research*, vol. 7, pp. 66-74, 2020, doi: 10.5152/ArcHealthSciRes.2020.535808.
- [9] D. Ijeri, P. Maidargi, R. Srinagar, "Traffic control system using image processing," *2020 IEEE Bangalore Humanitarian Technology Conference (B-HTC)*, pp. 1-6, 2020, doi: 10.1109/BHTC50970.2020.9298014.
- [10] J. Martínez Sandoval, E. Bracamontes, M. Martinez-Rosas, M. Miranda, H. de Ávila, "Image Processing Applied in Agriculture," pp. 201-226, 2012, doi: 10.1201/b12298-9.

- [11] W. Sun, L. Li, "Design of process products based on image processing multimode interaction," *Scientific Programming*, 2021, doi: 10.1155/2021/9925764.
- [12] V. V. Shah, "Image processing and its military applications," *Defence Science Journal*, vol. 37, pp. 457-468, 1987.
- [13] J. Kapur, A. Baregar, "Security using image processing," *International Journal of Managing Information Technology*, vol. 5, pp. 13-21, 2013, doi: 10.5121/ijmit.2013.5202.
- [14] M. M. D. Savio, T. Deepa, A. Bonasu, T. S. Anurag, "Image processing for face recognition using haar, hog, and svm algorithms," *Journal of Physics: Conference Series*, vol. 1964, no. 6, 2021, doi:10.1088/1742-6596/1964/6/062023.
- [15] Z. Küçükkara, A. E. Tümer, "An image processing oriented optical mark recognition and evaluation system," *International Journal of Applied Mathematics, Electronics and Computers*, vol. 6, pp. 59-64, 2018.
- [16] P. van der Stelt, "From pixel to image analysis," *Dentomaxillofacial Radiology*, vol. 50, no. 2, 2021, doi: 10.1259/dmfr.20200305.
- [17] Y. Li, Y. Tu, J. Lu, Y. Wang, "A security transmission and storage solution about sensing image for blockchain in the internet of things," *Sensors*, vol. 20, no. 3, p. 916, Feb. 2020, doi: 10.3390/s20030916.
- [18] K. Koptyra, M. R. Ogiela, "Imagechain—application of blockchain technology for images," *Sensors*, vol. 21, no. 1, p. 82, Jan. 2021, doi: 10.3390/s21010082.
- [19] W. Zhang, Q. Wang, M. Li, "Privacy-preserving collaborative training for medical image analysis based on multi-blockchain," *Combinatorial Chemistry High Throughput Screening*, vol. 23, no. 10, pp. 1025-1036, 2020, doi: 10.2174/1386207323666201022110616.
- [20] F. Liu, C. Y. Yang, Y. Jie, D. L. Kong, A. M. Zhou, J. Y. Qi, Z. B. Li, "A hybrid with distributed pooling blockchain protocol for image storage," *Scientific Reports*, vol. 12, no. 1, pp. 1-12, 2022, doi: 10.1038/s41598-022-07494-9.
- [21] D. Jeevanandham, R. Chinnasamy, T. Senthil, "Security and privacy of cloud- and iot-based medical image diagnosis using fuzzy convolutional neural network," *Computational Intelligence and Neuroscience*, 2021, doi: 10.1155/2021/6615411.

- [22] S. Walt, J. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. Warner, N. Yager, E. Gouillart, T. Yu, "scikit-image: Image processing in python," 2014, doi: 10.7287/peerj.preprints.336v2.
- [23] A. Khanna, A. Sah, V. Bolshev, A. Burgio, V. Panchenko, M. Jasinski, "Blockchain–cloud integration: A survey," *Sensors*, vol. 22, no. 15, pp. 5238, 2022, doi: 10.3390/s22145238.
- [24] G. Ünal, "Uluyol, Blok zinciri teknolojisi," *Bilişim Teknolojileri Dergisi*, cilt. 13, no. 2, ss. 167-175, 2020, doi: 10.17671/gazibtd.516990.
- [25] K. Carvalho, J. Granjal, "Security and privacy for mobile iot applications using blockchain," *Sensors*, vol. 21, no. 17, doi: 10.3390/s21175931.
- [26] A. Mehbodniya, R. Neware, D. S. Vyas, M. Kumar, P. S. D., S. Ray, "Blockchain and ipfs integrated framework in bilevel fog-cloud network for security and privacy of iomt devices," *Computational and Mathematical Methods in Medicine*, pp. 1-9, 2021, doi: 10.1155/2021/7727685.
- [27] D. Mishra, V. Bhattacharjee, S. Saket, S. Mishra, "Security provisions in smart edge computing devices using blockchain and machine learning algorithms: a novel approach," *Cluster Computing*, vol.27, no.1, pp.27-52, 2024, doi: 10.1007/s10586-022-03813-x.
- [28] J. Chen, X. Chen, C. L. Chen, "A traceable blockchain-based vaccination record storage and sharing system," *Journal of Healthcare Engineering*, 2022, doi: 10.1155/2022/2211065.
- [29] V. Wylde, N. Rawindaran, J. Lawrence, R. Balasubramanian, E. Prakash, A. Jayal, I. Khan, C. Hewage, J. Platts, "Cybersecurity, data privacy and blockchain: A review," *SN Computer Science*, vol. 3, no.2, 2022, doi: 10.1007/s42979-022-01020-4.
- [30] A. Manawar, "An innovative and secure platform for leveraging the blockchain approach for online exams," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 5, pp. 99-108, 2023, doi: 10.34306/att.v5i1.314.
- [31] R. Sattar, M. T. B. Hossain Efty, T. Rafa, T. Das, M. S. Samad, A. Pathak, M. Khandaker, M. H. Ullah, "An advanced and secure framework for conducting

online examination using blockchain method," *Cyber Security and Applications*, vol. 1, 2022, doi: 10.1016/j.csa.2022.100005.

- [32] S. Aishwarya, S. Ramya, S. Subhiksha, S. Samundeswari, "Detection of impersonation in online examinations using blockchain," *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, pp. 1-5, 2020, doi: 10.1109/ICPECTS49113.2020.9337001.
- [33] D.-M. Nguyen, Q.-H. Luu, N. Huynh-Tuong and H.-A. Pham, "MB-PBA: Leveraging Merkle Tree and Blockchain to Enhance User Profile-based Authentication in E-Learning Systems," *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, Ho Chi Minh City, Vietnam, pp. 392-397, 2019, doi: 10.1109/ISCIT.2019.8905114.
- [34] M. Zhao, W. Liu, A. N. M. Saif, B. Wang, R. A. Rupa, K. M. A. Islam, S. M. M. Rahman, N. Hafiz, R. Mostafa, M. A. Rahman, "Blockchain in online learning: A systematic review and bibliographic visualization," *Sustainability*, vol. 15, no. 2, doi: 10.3390/su15021470.