

**T.C.
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
YÜKSEK LİSANS**

**ARKEOLOJİK ALANLARDA YAPAY ZEKA
DESTEKLİ ENVANTER TAKİP YAZILIMI:
ARKHESTOR**

Can YASTIOĞLU

Danışman: Dr. Öğr. Üyesi Ersan OKATAN

BURDUR, 2024

ETİK KURALLARA UYGUNLUK BEYANI

Burdur Mehmet Akif Ersoy Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili hükümleri uyarınca **Yüksek Lisans** Tezi olarak sunduğum “**Arkeolojik Alanlarda Yapay Zekâ Destekli Envanter Takip Yazılımı: ARKHESTOR**” başlıklı bu tezin;

- Kendi çalışmam olduğunu,
- Sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi,
- Bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi,
- Kullandığım verilerde değişiklik yapmadığımı,
- Tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını,
- Bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı,
- Bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı,

bildirir, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

06/08 /2024

(İmza)

Can YASTIOĞLU

ÖNSÖZ

Tez çalışmam boyunca rehberliği, değerli görüşleri ve geliştirici geribildirimleriyle katkılarını esirgemeyen, yol gösterici tavrı ile tezimin son halinin verilmesinde büyük katkıları olan danışmanım **Dr. Öğr. Üyesi Ersan OKATAN**'a sonsuz teşekkürlerimi sunarım.

Yine tez sürecimin başından sonuna kadar bilgi ve deneyimleri ile tez konuma yaptığı katkılardan dolayı değerli hocam Prof. Dr. Ali HAKAN IŞIK'a teşekkür ederim.

Ayrıca, tez konumu belirleme aşamasında sadece teorik katkı değil uygulama geliştirerek sahaya da katkı sunma fırsatı bulmamı sağlayan, konumun kullanılabilir bir uygulamaya dönüşmesinde fikirleri ve arkeoloji araştırmalarındaki bilgileri ile yol gösteren Prof. Dr. Feyzullah Eray DÖKÜ'ye de teşekkürlerimi sunarım.

Sadece tez sürecimde değil, çocukluğumdan bu yana her gün bir şekilde iletişim kurduğumuz, dertleştığımız ve güldüğümüz sevgili arkadaşım Öğr. Gör. Erkan KAYNAK'a yoldaşlığı için teşekkür ederim.

Son olarak tez çalışmam sırasında iş yoğunluğunun oldukça fazla olduğu bir dönemde dahi tezimin editörlüğünü ve okumasını yaparak bana destek olan sevgili eşim Doç. Dr. Seher YASTIOĞLU'na sonsuz teşekkürlerimi sunarım. Göstermiş olduğun özveri ve yardımların sayesinde bu süreci daha kolay ve verimli bir şekilde tamamlayabildim. Her zaman yanımda olduğun ve birlikte yürüdüğümüz yol için minnettarım.

Ağustos, 2024

Can YASTIOĞLU

İÇİNDEKİLER

	Sayfa
ÖNSÖZ	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
ÇİZELGE DİZİNİ.....	vii
SİMGELER KISALTMALAR DİZİNİ.....	viii
ÖZET	ix
1. GİRİŞ	1
2. GENEL BİLGİLER	3
2.1. Arkeolojik Veri Yönetimi ve Standartlar	4
2.1.1. CIDOC CRM (Conceptual Reference Model)	4
2.1.2. Dublin Core Metadata Initiative (DCMI)	4
2.1.3. Tekil Kontekst Kayıt Sistemi (Single Context Recording System).....	5
2.1.4. Stratigrafi ve Harris Matrisi.....	5
2.1.5. ISO 21127.....	5
2.2. Yapay Zekâ ile Özellik Çıkarımı ve Benzerlik Fonksiyonları	5
2.3. Arkeolojik Kazı Alanı Envanter ve Takip Yazılımları	6
3. MATERYAL ve YÖNTEM.....	8
3.1. Frontend.....	8
3.1.1. Tailwind CSS.....	9
3.1.2. Pinia	9
3.1.3. Vue.js	10
3.1.4. JSON Web Token (JWT).....	11
3.1.5. Kimlik Doğrulama	12
3.1.6. QR Kod.....	14
3.1.7. Harita Servisleri ile Veri Görselleştirme	15
3.2. Backend	16
3.2.1. PHP	16
3.2.2. Framework.....	17
3.2.3. Laravel	18
3.2.4. Veritabanı: MySQL	21
3.2.5. OpenCV	22
3.2.6. TensorFlow ve EfficientNetB0.....	22
3.2.7. Benzerlik Fonksiyonları ve EfficientNetB0, ResNet50, VGG16, InceptionV3 Ön Eğitimli Modeller ve Benzerlik Skorlarını Hesaplayan Servis.....	23
3.2.8. Yayınlama: Nginx.....	26
3.3. Sistem Mimarisi	26
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	29
4.1.1. Kullanıcı Girişi (Karşılama)	29
4.1.2. Tanımlamaların Yapılması	31
4.1.3. Sezon Tanımlanması.....	31
4.1.4. Birim Türlerinin Tanımlanması.....	32
4.1.5. Buluntu Türlerinin Tanımlanması	34
4.1.6. Kullanıcıların ve Rollerin Tanımlanması	35
4.1.7. Takımların Tanımlanması ve Yönetimi.....	37
4.1.8. Birim Şemalarının Tanımlanması ve Yönetimi.....	39
4.1.9. Birimlere Yetkili veya Takımların Tanımlanması.....	43

4.1.10. Yetkili Kişilerin Tanımlanması;	43
4.1.11. Takımların Tanımlanması	44
4.1.12. Kullanıcı Arayüzü: Yetkilendirilmiş Kullanıcılar için İşlem Menüsü	45
4.1.13. Buluntu Ekleme Menüsü	45
4.1.14. Buluntu Arama	48
4.1.15. QR Kod ile Arama	48
4.1.16. Özelliklere Göre Arama	49
4.1.17. Yapay Zekâ ile Buluntu Arama	50
4.1.18. Buluntu Geçmişi	51
4.1.19. Transfer İşlemleri	54
4.1.20. Ön Eğitilmiş Modeller ile Benzerlik Fonksiyonlarının Performans Kıyaslaması	55
5. SONUÇ VE ÖNERİLER	82
6. KAYNAKLAR	85



ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1. JSON WEB Token Yapısı	12
Şekil 3.2. Kimlik Doğrulama Akış Diyagramı	13
Şekil 3.3. Buluntu Lokasyonların Google Maps Üzerindeki Görünümü.....	16
Şekil 3.4. Framework Kullanıcı Deneyim Puanları	21
Şekil 4.1. Kullanıcı Giriş Ekranı.....	30
Şekil 4.2. Yönetici İşlemleri Menüsü	31
Şekil 4.3. Kazı Sezonlarının Tanımlanması.....	32
Şekil 4.4. Birim Türlerinin Tanımlanması ve Birim Yapılarının Oluşturulması.....	33
Şekil 4.5. Kullanıcı Tanımlama Ekranı	35
Şekil 4.6. Takım Tanımlama ve Takım Yönetimi Ekranı.....	38
Şekil 4.7. Kazı Alanı İşleyiş Modülleri	40
Şekil 4.8. Birim Şemalarının Tanımlanması ve Yönetilmesi Ekranı.....	41
Şekil 4.9. Birim Şemasına Yeni Yapı Ekleme Ekranı	42
Şekil 4.10. Yöneticiler için Birim Kullanıcısı Tanımlama Ekranı.....	43
Şekil 4.11. Takımların Birimlere Atanma Ekranı.....	44
Şekil 4.12. Kullanıcı Arayüz Ekranı	45
Şekil 4.13. Buluntu Yeri Kayıt	46
Şekil 4.14. Buluntu Durumu Kaydı	46
Şekil 4.15. Buluntu Türü Kaydı.....	46
Şekil 4.16. Buluntu Rengi Kaydı	47
Şekil 4.17. Buluntuya Dair Detay Bilgilerin Kaydı.....	47
Şekil 4.18. QR ile Buluntu Arama	49
Şekil 4.19. Özelliklere göre Buluntu Arama Ekranı	49
Şekil 4.20. Yapay Zekâ Destekli Buluntu Arama Ekranı	51
Şekil 4.21. Buluntu Geçmişi Ekranı	52
Şekil 4.22. Buluntu Üzerinde Yapılan İşlemler Ekranı	53
Şekil 4.23. Buluntu Yeni İşlem Ekranı	53
Şekil 4.24. Transfer İşlemleri Ekranı.....	54
Şekil 4.25. EfficientNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Cosine).....	56

Şekil 4.26. EfficientNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Euclidean).....	57
Şekil 4.27. EfficientNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Jaccard).....	58
Şekil 4.28. EfficientNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Overlap).....	59
Şekil 4.29. EfficientNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Pearson).....	60
Şekil 4.30. InceptionV3 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Cosine).....	61
Şekil 4.31. InceptionV3 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Euclidean).....	62
Şekil 4.32. InceptionV3 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Jaccard).....	63
Şekil 4.33. InceptionV3 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Overlap).....	64
Şekil 4.34. InceptionV3 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Pearson).....	65
Şekil 4.35. ResNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Cosine).....	66
Şekil 4.36. ResNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Euclidean).....	67
Şekil 4.37. ResNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Jaccard).....	68
Şekil 4.38. ResNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Overlap).....	69
Şekil 4.39. ResNet Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Pearson).....	70
Şekil 4.40. VGG16 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Cosine).....	71
Şekil 4.41. VGG16 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Euclidean).....	72
Şekil 4.42. VGG16 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Jaccard).....	73

Şekil 4.43. VGG16 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Overlap).....	74
Şekil 4.44. VGG16 Modellerinin Farklı Eşik Değerlerdeki ile Performans Sonuçları (Pearson).....	75
Şekil 4.45. Farklı Eşik Değerlerindeki Model ve Benzerlik Fonksiyonlarının Accuracy Değerlerinin Kıyaslanması	76
Şekil 4.46. Farklı Eşik Değerlerindeki Model ve Benzerlik Fonksiyonlarının Precision Değerlerinin Kıyaslanması	77
Şekil 4.47. Farklı Eşik Değerlerindeki Model ve Benzerlik Fonksiyonlarının Recall Değerlerinin Kıyaslanması	78
Şekil 4.48. Farklı Eşik Değerlerindeki Model ve Benzerlik Fonksiyonlarının F1 Skorlarının Kıyaslanması.....	79

ÇİZELGE DİZİNİ

Sayfa

Çizelge 3.1. PHP Frameworklerinin Güçlü ve Zayıf Yönleri.....	20
---	----



SİMGELER KISALTMALAR DİZİNİ

ARKHESTOR	: ARKESTHOR Uygulaması
API	: Application Programming Interface
CSS	: Cascading Style Sheets
HTML	: Hypertext Markup Language
JSX	: JavaScript XML
JWT	: JSON Web Token
MySQL	: My Structured Query Language
Nginx	: Engine-X (Web Sunucusu)
OOP	: Object-Oriented Programming
OpenCV	: Open Source Computer Vision Library
PHP	: Hypertext Preprocessor
Pinia	: State Management Library for Vue.js
QR Kod	: Quick Response Code
SFC	: Single-File Component
SQL	: Structured Query Language
STC	: Single Threaded Core
VGG16	: Visual Geometry Group 16-Layer Network
Vue.js	: A Progressive JavaScript Framework

ÖZET

Yüksek Lisans Tezi

**Arkeolojik Alanlarda Yapay Zekâ Destekli Envanter Takip Yazılımı:
ARKHESTOR**

Can Yastıoğlu

**Burdur Mehmet Akif Ersoy Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı**

Danışman: Dr. Öğr. Üyesi Ersan OKATAN

Ağustos, 2024

Bu tez çalışmasında, arkeolojik buluntuların dijital olarak kayıt altına alınması, sınıflandırılması, analiz edilmesi ve raporlanması için bir yazılım sistemi geliştirilmiştir. Sistem, arkeolojik kazı alanlarında buluntuların kaydedilmesi, depolanması, tasnif edilmesi, raporlanması ve akademik çalışmalar için analiz edilmesi süreçlerini verimli hale getirmeyi amaçlamaktadır. Yazılımın geliştirilmesi sürecinde; ön yüz geliştirme (frontend) için Vuejs, arka yüz (backend) için Laravel, veri tabanı olarak MySQL kullanılmıştır. Ayrıca benzer buluntuların tespiti için Python dili ve OpenCv, TensorFlow gibi kütüphanelerle makine öğrenmesi ve görüntü işleme teknikleri kullanılarak, Cosine Similarity benzerlik fonksiyonu ve EfficientNetB0 modeli seçilmiştir. Bu modelin, buluntuların özelliklerini çıkarma ve veritabanına kaydetme sürecinde yüksek doğruluk sağladığı gözlemlenmiştir. Cosine Similarity yöntemiyle yapılan analizlerde %92 doğruluk oranı elde edilmiştir. Bunun yanında geliştirilen yazılım, QR kod ve barkod teknolojileriyle buluntuların kaydını ve takibini kolaylaştırmaktadır. Sonuç olarak, geliştirilen yazılımın arkeologlara ve araştırmacılara kazı alanında buluntuların kayıt altına alınmasından, tasniflemeye ve raporlamaya kadar birçok alanda zaman kazandıracağı ve bilimsel analizleri hızlandıracağı düşünülmektedir.

Anahtar Kelimeler: arkeolojik buluntu envanter kayıt yazılımı, arkeolojik araştırma veri yönetim yazılımı, tek sayfa uygulaması, laravel çatısı, ön eğitilmiş model, benzerlik fonksiyonları



ABSTRACT

M. Sc. Thesis

**Artificial Intelligence Supported Inventory Tracking Software for Archaeological
Sites: ARKHESTOR**

Can YASTIOGLU

**Burdur Mehmet Akif Ersoy University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering Science**

Supervisor: Asst. Prof. Dr. Ersan OKATAN

August, 2024

The objective of this thesis was to develop a software system for the digital recording, classification, analysis, and reporting of archaeological findings. The system's objective is to facilitate the optimization of processes associated with the recording, storage, classification, reporting, and analysis of findings from archaeological excavation sites for academic research. In the course of developing the software, Vue.js was utilized for front-end development, Laravel for back-end development, and MySQL as the database. Furthermore, for the identification of analogous findings, machine learning and image processing techniques were employed using Python and libraries such as OpenCV and TensorFlow, with the Cosine Similarity function and the EfficientNetB0 model being selected. It was observed that this model provided high accuracy in the process of feature extraction and database recording of the findings. Analyses conducted using the Cosine Similarity method achieved a 92% accuracy rate. Furthermore, the developed software facilitates the recording and tracking of findings through the use of QR code and barcode technologies. In conclusion, the developed software saves time for archaeologists and researchers in various areas, from the recording of findings in excavation sites to classification and reporting, thereby accelerating scientific analyses.

Keywords: Archaeological inventory and research data management software, single page application, laravel framework, pretrained model, similarity functions

1. GİRİŞ

Arkeolojik kazı alanlarında keşfedilen buluntuların kayıt altına alınması, sınıflandırılması, analiz edilmesi ve raporlanması süreçleri, arkeoloji biliminin en temel ve kritik aşamalarından birini oluşturmaktadır. Geleneksel yöntemlerle yürütülen envanter yönetimi, genellikle manuel ve kâğıt tabanlı sistemlere dayanmaktadır. Bu yöntemler, zaman alıcı ve hata yapmaya müsait olmasının yanı sıra, verilerin sistematik bir şekilde organize edilmesini ve erişilebilirliğini zorlaştırmaktadır. Bunun yanında, arkeologların sahada topladıkları verilerin etkin bir şekilde işlenmesi, buluntuların güvenli bir şekilde saklanması ve bu verilerin akademik araştırmalarda ve restorasyon çalışmalarında kullanılmak üzere hızlıca erişilebilir olması büyük önem taşımaktadır. Buradan hareketle, arkeolojik buluntuların dijital bir platform üzerinden kayıt altına alınması ve yönetilmesinin, modern arkeoloji çalışmalarının verimliliğini ve doğruluğunu artırmak için önemli bir gereklilik haline gelmiş durumdadır.

Bu çalışmanın temel amacı, arkeolojik kazı alanlarında keşfedilen buluntuların dijital olarak kayıt altına alınması, sınıflandırılması, analiz edilmesi ve raporlanmasını sağlayacak bir yazılım sistemi geliştirmektir. Geliştirilecek sistem, QR kod ve barkod teknolojileri kullanarak buluntuların hızlı ve doğru bir şekilde kaydedilmesini ve izlenmesini mümkün kılmalıdır. Bu dijital platform, buluntuların depolanması, kategorize edilmesi ve akademik çalışmalar için gerekli analizlerin yapılması, güvenli bir şekilde aktarılabilmesi, depolanabilmesi ve restorasyon süreçlerini verimli hale getirmeyi amaçlamaktadır. Böylece özellikle, buluntuların kaydedilmesi ve analiz edilmesi sırasında ortaya çıkan koordinasyon eksiklikleri, verimsiz depolama süreçleri ve veritabanlarına erişim zorlukları gibi sorunların önüne geçilmesi hedeflenmektedir.

Bu tez kapsamında geliştirilen sistemde, makine öğrenmesi ve görüntü işleme teknikleri kullanılmıştır. Buluntu görüntülerine ait görsel özelliklerin detaylı bir şekilde analiz edilebilmesi için Python dili ile OpenCv ve TensorFlow gibi kütüphaneler kullanılmıştır. Bu amaçla, ön eğitilmiş EfficientNetB0, ResNet50, VGG16 ve InceptionV3 gibi modeller üzerinde testler yapılmış ve performans karşılaştırmaları gerçekleştirilmiştir. EfficientNetB0 modeli, yüksek doğruluk ve dengeli performans sunmasının yanı sıra daha az hesaplama gücü gerektirerek kaynakları verimli kullanması nedeniyle tercih edilmiştir.

Söz konusu süreçler, buluntuların dijital platformda hızlı bir şekilde işlenmesini ve yeni eklenen buluntuların veritabanına eklenmesini önemli ölçüde hızlandırmaktadır.

Benzerlik analizi yapan Flask tabanlı servis, belirli bir periyotta bir veritabanını kontrol ederek yeni eklenen buluntuları tespit etmeli ve bu buluntuların özelliklerini çıkararak veritabanındaki mevcut buluntularla karşılaştırmaktadır. Bu yaklaşım, sistemin sürekli güncel kalmasını ve verilerin doğruluğunu sağlayacaktır.

Geliştirilen sistem ayrıca, kullanıcı dostu bir arayüz sunarak, kullanıcıların buluntulara kolayca erişebilmesini sağlayacaktır. Geliştirilen envanter yazılımında kullanıcılar, buluntuların QR kodlarını taratarak veya barkod numaralarını girerek, bu buluntular hakkında detaylı bilgilere ve yapılan işlemlere hızlıca ulaşabileceklerdir. Harita ve coğrafi bilgi sistemleri (GIS) entegrasyonu ile buluntuların coğrafi konumları harita üzerinde görselleştirilecek, bu da kullanıcıların buluntuların dağılımını ve coğrafi ilişkilerini daha iyi anlamalarına yardımcı olacaktır. Böylece arkeologlar ve araştırmacılar, buluntular arasında hızlı ve etkili bir şekilde benzerlik analizi yapabilecek, bu sayede zaman kazandıran ve bilimsel araştırmaların kalitesini artıran bir süreç oluşturulmuş olacaktır.

Sonuç olarak, bu çalışma ile arkeolojik buluntuların dijital yönetimi ve analizi süreçlerinin modernize edilmesi ve daha verimli hale getirilmesi hedeflenmektedir. Geliştirilen yazılımın, arkeologlara ve araştırmacılara önemli ölçüde zaman kazandıracağı, bilimsel araştırmaların kalitesini artıracığı ve arkeolojik verilerin daha etkin kullanılmasını sağlayacağı beklenmektedir. Ayrıca, bu yazılım, arkeolojik verilerin korunması ve geleceğe aktarılması açısından da büyük bir öneme sahiptir. Dijital platformların sağladığı imkanlar, arkeolojik buluntuların daha geniş bir kitleye erişebilmesini ve bilimsel verilerin daha güvenilir bir şekilde paylaşılmasını mümkün kılacaktır.

2. GENEL BİLGİLER

Bu bölümde, arkeolojik veri yönetimi ve standartlarından başlayarak, kazı alanlarında kullanılan envanter ve takip yazılımlarına kadar geniş bir yelpazede bilgi sunulmuştur. Arkeolojik verilerin güvenilir ve sürdürülebilir bir şekilde yönetilmesi, uluslararası standartların benimsenmesiyle mümkündür. CIDOC CRM ve Dublin Core Metadata Initiative gibi standartlar, bu verilerin sistematik olarak toplanmasını ve işlenmesini sağlarken, Single Context Recording System ve Harris Matrisi gibi yöntemler, arkeolojik buluntuların mekânsal ve stratigrafik bağamlarını doğru bir şekilde kaydetmeye olanak tanımaktadır. Bu yöntemler ve standartlar, arkeolojik çalışmaların bilimsel doğruluğunu artırarak, bulguların gelecekteki araştırmalar için güvenilir bir şekilde saklanmasını ve erişilmesini garanti etmektedir. Bununla birlikte, yapay zekâ ve derin öğrenme teknolojileri, arkeolojik buluntuların analizinde büyük avantajlar sunmaktadır. Burada, EfficientNetB0 gibi ön eğitilmiş modeller, görüntü tabanlı verilerin işlenmesi ve sınıflandırılmasında yüksek doğruluk oranlarıyla öne çıkmaktadır. Cosine Similarity, Jaccard, Overlap ve Euclidean gibi benzerlik fonksiyonları ise, buluntular arasındaki benzerliklerin nicel olarak değerlendirilmesini sağlamaktadır. Bu fonksiyonların kullanımı, arkeologların veri yoğun çalışmalarını daha etkin bir şekilde yönetmelerine yardımcı olmakta ve büyük veri setleri içindeki gizli kalmış ilişkileri ortaya çıkarmaktadır.

Arkeolojik buluntuların kayıt altına alınması, kazı ve araştırma sürecinin kritik bir parçasını oluşturmaktadır. Bu süreçte, saha notlarını tutmak, buluntuların yerini, bağlamını ve ilgili bilgileri kaydetmek, buluntuların gelecekteki analizler için doğru ve ayrıntılı bir şekilde belgelenmesini sağlamaktadır. Buluntuların kayıt altına alınması, belgelenmesi, sınıflandırılması, arşivlenmesi ve korunması için geleneksel yazı ile tutulan ve genellikle fiziksel olarak arşivlenen kayıtların yanında dijital veritabanlarının da kullanılmaya ve önerilmeye başladığı görülmektedir. Dijital kayıt sistemleri kazı alanında yer alan buluntulara dair kayıtların arkeolojik bağlamı, materyallerin özellikleri ve buluntuların konumu gibi önemli bilgileri içermekle birlikte; buluntulara dair bilgilerin daha sonra yeniden gözden geçirilmesi, kolayca düzenlenmesi, görsellerin farklı versiyonlarının takip edilmesi, benzer buluntularla ilişkilendirme, hızlı şekilde raporlama gibi daha fazla detay, erişilebilirlik ve esneklik sunmaktadır. Dijital envanter sistemlerinin sunmuş olduğu avantajlar temelinde ve aşağıda sunulan başlıklar ışığında,

bu tez kapsamında ARKHESTOR yazılımı geliştirilerek, arkeolojik buluntuların dijital ortamda en yüksek doğruluk ve verimlilikte yönetilmesini amaçlanmıştır.

2.1 Arkeolojik Veri Yönetimi ve Standartlar

Arkeolojik verilerin toplanması, yönetilmesi ve analiz edilmesinin, arkeolojik araştırmaların başarısı için kritik öneme sahip olduğu görüşü hakimdir. Bu verilerin güvenilirliği ve sürdürülebilirliği, uluslararası kabul görmüş CIDOC CRM (Conceptual Reference Model), Dublin Core Metadata Initiative (DCMI), Tekil Kontekst Kayıt Sistemi (Single Context Recording System), Stratigrafi ve Harris Matrisi, ISO 21127 standartlar ve metodolojiler aracılığıyla sağlanmaktadır. Bu standartlar, arkeolojik verilerin bütüncül bir yaklaşımla ele alınmasını ve bu verilerin gelecek nesillere doğru ve eksiksiz bir şekilde aktarılmasını sağlayarak, arkeoloji biliminin ilerlemesine önemli katkılar sunmaktadır.

Ayrıca her bir standart, belirli bir veri yönetimi ihtiyacını karşılamak üzere tasarlanmış olup, verilerin disiplinler arası çalışmalarda kullanılabilirliğini artırmak ve uzun vadeli korunmasını sağlamak için kritik rol oynamaktadır. Arkeolojik araştırmaların uluslararası düzeyde tutarlı ve güvenilir olmasını sağlamak için bu standartların benimsenmesi ve uygulanması büyük önem arz etmektedir.

2.1.1 CIDOC CRM (Conceptual Reference Model)

Bu standartlar arasında öne çıkan CIDOC CRM (Conceptual Reference Model), kültürel miras alanındaki bilgi yönetimi için geliştirilmiş bir kavramsal modeldir (Henninger, 2018a). Uluslararası Müzeler Konseyi'nin dokümantasyon komitesi tarafından geliştirilen bu model, arkeolojik verilerin tanımlanması, sınıflandırılması ve ilişkilendirilmesi konularında bir çerçeve sunmaktadır. Özellikle arkeolojik buluntuların kontekstini ve aralarındaki ilişkileri standart bir formatta kaydetmek için kullanılan CIDOC CRM, farklı veri setlerinin entegrasyonunu sağlayarak geniş kapsamlı veri analizi ve bilgi paylaşımını mümkün kılmaktadır.

2.1.2 Dublin Core Metadata Initiative (DCMI)

Arkeoloji de dahil olmak üzere çeşitli alanlarda metadatanın standartlaştırılması ve yönetilmesi konusunda önemli bir rol oynamaktadır. DCMI, dijital kaynakların tanımlanmasında yaygın olarak kullanılan 15 temel metadata unsurunu belirlemiştir (Henshaw ve Valauskas, 2001). Bu unsurlar, başlık, yaratıcısı, konu, açıklama,

yayımlayıcı, katkıda bulunan, tarih, tür, format, tanımlayıcı, kaynak, dil, ilişki, kapsama alanı ve haklar gibi önemli bilgileri kapsamaktadır. Bu standartlaştırılmış metadata unsurlarına uyulması, arkeolojik verilerin tanımlanması ve erişiminde tutarlılık ve birlikte çalışabilirlik sağlamaktadır.

2.1.3 Tekil Kontekst Kayıt Sistemi (Single Context Recording System)

Tekil Kontekst Kayıt Sistemi (Single Context Recording System), arkeolojik birimlere benzersiz kontekst numaraları atanmasını vurgulayarak kazı sırasında mekânsal ilişkilerin ve stratigrafik dizilimlerin sistematik bir şekilde belgelenmesini sağlamaktadır (Roosevelt vd., 2015). Bu yapılandırılmış yaklaşım, her bir birimin tam yerinin ve stratigrafik bağlamının titizlikle kaydedilmesini garanti altına alınmasını sağlamakta ve bu sayede araştırmacıların sitenin tarihini yeniden yapılandırmasına ve arkeolojik bulguları iyi tanımlanmış bir çerçeve içinde yorumlamasına olanak tanımaktadır. Bu sistematik kayıt süreci, veri bütünlüğünü artırmanın yanı sıra arkeolojik buluntuların analizini ve yorumlanmasını da kolaylaştırmaktadır.

2.1.4 Stratigrafi ve Harris Matrisi

Stratigrafi ve Harris Matrisi, arkeolojik buluntuların kronolojik sırasını ve stratigrafik ilişkilerini belirlemek için kullanılan temel yaklaşımlardan biridir (Henninger, 2018b). Stratigrafi, arkeolojik buluntuların zaman içinde biriktikleri katmanları inceleyerek bu katmanlar arasındaki ilişkileri belirlerken, Harris Matrisi ise bu ilişkilerin görsel bir temsilini sağlayarak kazı alanının tarihsel gelişimini anlamak için kritik bir araç sunmaktadır.

2.1.5 ISO 21127

Kültürel miras bilgisi değişimi için geliştirilmiş uluslararası bir standart olup, arkeolojik verilerin bilgisayar sistemleri arasında nasıl paylaşılacağına dair bir çerçeve sunmaktadır (Henninger, 2018b). Bu standart, özellikle dijital arşivleme ve veri paylaşımı alanında önemli bir rol oynamakta olup, arkeolojik verilerin farklı sistemler arasında uyumlu bir şekilde transfer edilmesini sağlamaktadır.

2.2 Yapay Zekâ ile Özellik Çıkarımı ve Benzerlik Fonksiyonları

Yapay zekâ ve derin öğrenme teknolojileri, arkeolojik buluntuların analizinde önemli avantajlar sunmaktadır. Derin öğrenme modelleri, özellikle görüntü tabanlı

verilerin işlenmesi ve sınıflandırılmasında yüksek doğruluk oranları sağlamaktadır. Ön eğitilmiş modeller, örneğin EfficientNetB0, karmaşık verilerde özellik çıkarımı için güçlü araçlar sunarak arkeolojik buluntuların detaylı analizinde kritik bir rol oynamaktadır. Bu modeller, buluntuların görsel özelliklerini analiz ederek benzer buluntularla karşılaştırılmasına olanak tanımaktadır (Doğan ve Türkoğlu, 2019). Benzerlik fonksiyonları olarak kullanılan Cosine Similarity, Jaccard, Overlap ve Euclidean gibi yöntemler, buluntular arasındaki benzerlikleri nicel olarak değerlendirmektedir. Özellikle Cosine Similarity, çok boyutlu veri uzaylarında iki nesne arasındaki açısal benzerliği hesaplayarak arkeolojik verilerde yüksek doğruluk oranları elde edilmesine olanak tanımaktadır. Bu benzerlik fonksiyonlarının kullanımı, arkeologların veri yoğun çalışmalarını daha etkin bir şekilde yönetmelerine yardımcı olmakta ve büyük veri setleri içindeki gizli ilişkileri ortaya çıkarmaktadır (Bütüner vd., 2023). Yapay zekâ ile özellik çıkarımı ve benzerlik analizi, arkeolojik verilerin dijital platformlarda daha verimli bir şekilde işlenmesini ve değerlendirilmesini sağlamaktadır (Ülker ve Çamlı, 2023). Bu teknolojiler, arkeologlara buluntular arasındaki ilişkileri daha iyi anlama ve araştırma süreçlerini geliştirme imkânı sunmaktadır. Bu bağlamda, derin öğrenme modelleri ve benzerlik fonksiyonlarıyla desteklenen yapay zekâ teknolojileri, arkeolojik buluntuların analizinde önemli bir rol oynamakta ve arkeologlara daha kapsamlı, verimli ve doğru sonuçlar elde etme imkânı sunmaktadır.

2.3 Arkeolojik Kazı Alanı Envanter ve Takip Yazılımları

Arkeolojik kazı alanları envanter yazılımı, arkeologlar için kazı sırasında bulunan eserlerin kaydedilmesi, belgelenmesi ve yönetilmesi açısından büyük bir öneme sahiptir. Bu yazılım, kazı alanındaki envanterin düzenli tutulmasını sağlayarak arkeologlara çalışmalarında rehberlik eder. Arkeolojik kazı alanlarından elde edilen buluntuların envanterinin doğru bir şekilde yapılması, gelecekteki araştırmacıların bu verilere erişimini kolaylaştırmakta, bunun yanında kayıt sistematik bir kayıt altına alma yoluyla arkeolojik mirasın korunmasına da katkı sağlamaktadır.

Arkeolojik araştırmalarda, sahadan elde edilen verilerin toplanması, saklanması, analiz edilmesi ve paylaşılması süreçlerinde karşılaşılan zorluklar, veri yönetim sistemlerinin önemini her geçen gün arttırmaktadır. Bu bağlamda, Gidding vd. (2013) arkeoloji alanında veri yönetimi ihtiyaçlarını ele almışlardır. Bu ihtiyaçlar arasında, arkeolojik buluntuların doğru ve sistematik bir şekilde kayıt altına alınması, veri yoğun araştırma tekniklerinin uygulanabilirliğini artırmak için verilerin merkezi bir depo

aracılığıyla yönetilmesi ve bu verilerin görselleştirme ve analiz araçlarına kolayca entegre edilebileceği bir yazılımın geliştirilmesi yer almaktadır. Geleneksel yöntemleri bir tarafa bırakarak (ör. excel dosyasına kaydetme bu dosya üzerinden izlemelerin yapılması, yazılı belge oluşturma) bu ihtiyaçlara cevap verecek web tabanlı envanter ve kayıt sisteminin oluşturulması, böylece arkeoloji alanında teşhis ve analitik tekniklerin geniş çapta benimsenmesi hem araştırma hem de web tabanlı küratörlük ve veri yönetim sistemi için altyapı oluşturulmasını sağlamaktadır (Gidding vd., 2013).

Arkeolojik kazı alanları, geçmiş uygarlıklar ve toplumlar hakkında önemli bilgiler içermekte olup, bu bilgilerin korunması ve günümüze aktarılması büyük bir önem taşımaktadır. Arkeolojik kazı alanlarına ait envanterlerin doğru bir şekilde tutulması, kazı çalışmalarının verimliliğini artırabilmekte ve elde edilen bulguların korunmasını sağlayabilmektedir. Dolayısıyla, arkeolojik kazı alanlarında yapılan çalışmaların envanter yönetimi büyük bir titizlikle yürütülmesi ihtiyacına dikkat çekilmektedir (Keskin ve Tanaç Zeren, 2022).

Envanter kayıtlarının doğruluğu ve detayı, arkeolojik buluntuların analiz edilmesi, tarihlendirilmesi ve korunmasında kritik bir rol oynamaktadır. Özellikle, buluntu yerleri hakkında ayrıntılı bilgilerin envanter kayıtlarında yer alması, gelecekte yapılacak araştırmalar için büyük önem taşımaktadır (Özcan, 2024).

Arkeolojik kazı alanlarında yapılan çalışmaların envanter yönetimi, kazı sonuçlarının doğru bir şekilde belgelenmesini ve gelecek nesillere aktarılmasını sağlamaktadır. Bu sayede, arkeologlar ve tarihçiler, geçmiş uygarlıklar hakkında daha fazla bilgi edinebilmekte ve kültürel mirasın korunmasına katkıda bulunabilmektedirler (Tezer Altay vd., 2022). Bunun yanında envanter sistemin, kazı buluntularının kaybolmasını önlemekte ve arkeologlara buluntuları analiz etme ve yorumlama imkânı sunmakta; envanter kayıtlarının düzenli tutulması, arkeolojik çalışmaların sürekliliğini ve verimliliğini de artırabilmektedir (De La Torre vd., 2015).

Dolayısıyla, arkeolojik kazı alanları envanter yazılımlarının geliştirilmesi ve kullanılması, kazı çalışmalarının daha etkili bir şekilde yönetilmesine ve arkeologların verimliliğinin artmasına yardımcı olabilir. Bu yazılımlar, kazı alanlarındaki buluntuların detaylı bir şekilde belgelenmesini ve korunmasını sağlayarak arkeolojik araştırmalara büyük katkı sağlayabilir (Efkleidou vd., 2022).

3. MATERYAL ve YÖNTEM

Bu çalışma, arkeolojik buluntuların dijital envanter yönetim sistemi tarafından kayıt altına alınması, buluntular üzerinde yapılan tüm işlemlerin kaydedilmesi, ekiplerin oluşturulması, sistem yöneticilerinin tanımlanması, kazı alanı ağacının belirlenmesi, raporlama, arama, benzerlik analizlerinin yapılması ve kullanıcı dostu bir arayüz sunulması üzerine odaklanmaktadır.

Çalışmanın odağı doğrultusunda Materyal ve Yöntem bölümünde sırası ile; frontend için Vue.js, Tailwind, Pinia, backend için Laravel, benzer buluntuların tespiti için kosinüs benzerlik algoritması kullanarak geliştirilen servis için Python ve Flask, verilerin saklanması için MySQL ve yazılımın yayınlanması için Nginx gibi çeşitli teknolojilerin, yazılım dillerinin ve algoritmaların neden tercih edildiği, yazılım sisteminin yapılandırılması ve işlevselliği üzerindeki rolü açıklanmıştır.

3.1 Frontend

Tek sayfa web siteleri ve tek sayfa uygulama (single page application - SPA) geliştirme, dinamik ve etkileşimli kullanıcı arayüzleri oluşturmak için Vue.js, React.js, Angular ve JavaScript gibi yaygın olarak kullanılan popüler kütüphaneler ve çerçeveler, web geliştirme alanında birçok avantaj sunarak kullanıcı deneyimini artırmaya, performansı iyileştirmeye ve gezinmeyi daha akıcı hale getirmeye odaklanmaktadır (Steven vd., 2023). Ayrıca, kütüphane seçimi, hem ön uç hem de arka uç unsurlarını kapsayarak, web sitesinin veya uygulamanın genel kullanıcı deneyimine ve operasyonel verimliliğine katkıda bulunmaktadır (Gupta vd., 2024).

Tek sayfa uygulamalar (SPA'lar), ağ bant genişliği ihtiyacını azaltma ve hızlı gezinme süreçleri sağlama yetenekleri nedeniyle oldukça popüler olarak kullanılmaktadır. SPA'lar, kullanıcı etkileşimlerini ve duyarlılığı artırmak için JavaScript kütüphanelerini ve JSON (JavaScript Object Notation) gibi asenkron veri işleme yöntemlerini kullanarak web sayfasının yenilenmesinin önüne geçerek kullanıcı deneyimini arttırmaktadır (Putra vd., 2021).

Bunun yanında ön uç geliştiricileri sadece kullanıcının göreceği arayüzü geliştirmekten sorumludur. Bu durum, kullanıcı beklentilerine ve iş gereksinimlerine uygun, görsel açıdan çekici ve işlevsel arayüzler oluşturmak için yoğunlaşabilmelerine arka uç ile ilgili endişe duymamalarını sağlamaktadır (Cazenave vd., 2011; Jiang vd., 2022).

Genel olarak, ön uç geliştirme, ilgi çekici ve kullanıcı dostu arayüzler oluşturmayı hedefleyen web geliştirmenin kritik bir yönüdür. Vue.js, React.js ve JavaScript gibi teknolojilerden yararlanarak, geliştiriciler, kesintisiz gezinme, artırılmış performans ve geliştirilmiş güvenlik özellikleri sunan etkileşimli web siteleri ve SPA'lar oluşturabilmektedir.

3.1.1 Tailwind CSS

Tailwind CSS kullanarak bir web arayüzü geliştirirken, stil vermeye yönelik yardımcı sınıf yaklaşımı sunan bir çerçeve kullanmanın faydalarını göz önünde bulundurulması gerekliliği söz konusudur. Tailwind CSS, geniş bir yelpazede önceden oluşturulmuş sınıflar sunarak geliştirme sürecinin hızlandırılmasını sağlamaktadır. Bu sınıflar, HTML öğelerine doğrudan uygulanabilmekte, bu da hızlı prototip oluşturma ve web bileşenlerinin verimli bir şekilde stil verilmesine olanak tanımaktadır (Akbulut ve Çare, 2011).

Tailwind CSS'in avantajlarından yararlanan geliştiriciler, web arayüzünün işlevselliği ve yapısına odaklanırken, çerçevenin geniş yardımcı sınıf seti aracılığıyla tasarımı kolayca özelleştirebilmektedirler. Bu yaklaşım, geliştirme aşamasında hem verimliliği arttırmayı hem de uygulamanın genelinde tutarlılık olmasını sağlamaktadır.

Sonuç olarak, bir web arayüzünün geliştirme sürecine Tailwind CSS'in dahil edilmesi, geliştiricilere stili düzenleme, verimliliği artırma ve duyarlılığı sağlama konusunda güçlü bir araç sunmaktadır. Bu sebeple modern ve kullanıcı dostu bir web uygulamasının oluşturulmasına katkıda bulunmasından dolayı geliştirilen uygulamanın tasarımında Tailwind CSS tercih edilmiştir.

3.1.2 Pinia

Pinia, Vue.js için state yönetimini sadeleştiren bir Vue.js kütüphanesidir (Lange vd., 2024). SPA'da, arayüz üzerindeki gerekli değişkenler Pinia Store'ye kolayca ve direk bağlanabilmektedir. Bu bağlanan değişkenler, Vue.js içinden doğrudan değiştirilerek, yapılan bu değişiklik hiçbir ek kod yazmaya ihtiyaç duymadan arayüze yansımakta ve arayüzün de güncellenmesini sağlamaktadır.

Pinia'nın yazılımcılar tarafından sıkça kullanılan alternatifi olan VUEX'e göre öne çıkan başlıca avantajları şunlardır:

- Pinia'nın API'si, Vuex'e kıyasla daha basit ve sezgiseldir. Bu, özellikle yeni başlayan geliştiriciler için durum yönetimini daha erişilebilir hale getirmektedir.
- Pinia, Vuex'te gerekli olan birçok gereksiz kodu ortadan kaldırarak, durum değişiklikleri arasında daha az kod yazmayı içermektedir.
- Pinia, birden fazla store oluşturmaya olanak tanımaktadır. Başka bir ifade ile her store kendi bağımsız durum, eylemler ve getter'lara sahip olabilmektedir. Bu sayede, büyük projelerde, bileşenler arasında modülerlik ve ayrıma gidilmesi kolaylaşmaktadır. Her bileşen, sadece ihtiyaç duyduğu store'u kendisine dahil edebilmekte ve kullanabilmektedir.

Söz konusu avantajlar sebebiyle ARKHESTOR yazılımının geliştirilmesinde Pinia kütüphanesinin kullanılması tercih edilmiştir.

3.1.3 Vue.js

Son araştırmalarda, Vue.js kullanarak SPA geliştirmenin önemli ölçüde dikkat çektiği görülmektedir. Model-View-ViewModel (MVVM) mimarisine dayanan bir JavaScript çerçevesi olan Vue.js, etkileşimli ve verimli ön uç uygulamaları oluşturmak için bir platform sunmaktadır. Bu sayede Vue.js kullanımı, SPA'ların geliştirilmesinde hızlı arayüzlerinin inşa edilmesini ve kullanıcı deneyimini artırmayı sağlamaktadır (Zou vd., 2022). Ayrıca, Vue.js'in Axios gibi çeşitli eklentilerle uyumluluğu, asenkron istekleri kolaylaştırarak uygulamaların işlevselliğinin daha da artırılmasını sağlamaktadır (Gan vd., 2022).

Herhangi bir SPA uygulamasında, Vue.js, React, Angular gibi SPA kütüphanelerinden herhangi birinin seçilmesi genellikle geliştiriciye veya geliştirici ekibinin tercihinin bırakılmaktadır. Bu frameworkler, genel olarak öne çıkan alternatifleri ile aynı işi benzer şekillerde yapsalarda bu projede geliştiricinin hangi framework'ü kullanacağına karar vermesinde aşağıda sunulan Vue.js'in öne çıkan özellikleri etkili olmuştur:

- *Öğrenme ve Kullanım Basitliği:* Vue.js'in öğrenme eğrisinin geliştiriciler tarafından daha kolay algılandığında dair yaygın bir kanı bulunmaktadır. Geliştiricilerin, Vue'nun sezgisel ve kullanıcı dostu yapısı sayesinde, geliştiricilerin Vue'nin kullanımına daha hızlı uyum sağlayabildiği düşünülmektedir. Vue.js'nin temel yapısı, HTML, CSS ve JavaScript bilgisi olan herkes için erişilebilir ve anlaşılır bir yapıdadır.

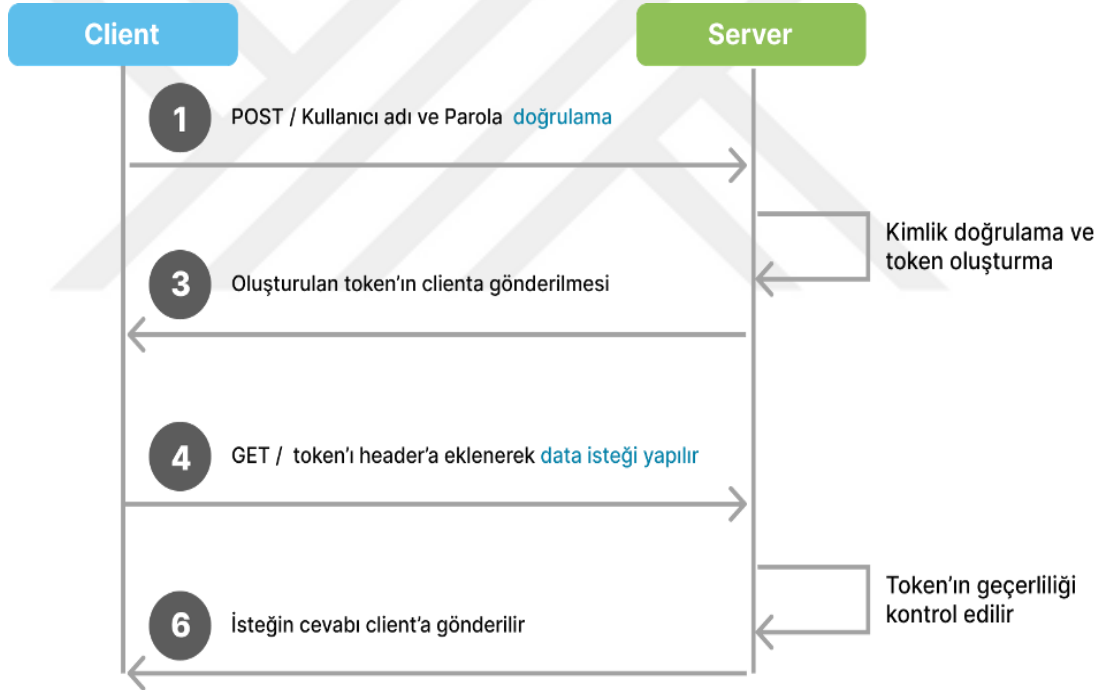
- *Detaylı Belgeler ve Dokümantasyon:* Vue.js'nin belgeleri oldukça kapsamlı ve anlaşılması kolaydır. Bu avantaj sayesinde, yeni başlayanların ve deneyimli geliştiricilerin hızlı bir şekilde sorunlarını çözmesine ve projelerine devam etmesine katkıda bulunmaktadır. Alternatifi olan Reactjs de kapsamlı belgelere sahip olmasına karşın, geliştiriciler tarafından Vue.js'nin belgeleri daha kullanıcı dostu olarak değerlendirilmektedir.
- *Komponent Tabanlı Mimari:* Vue.js, Reactjs gibi komponent tabanlı bir mimariyi desteklemekte, ancak Vue'nun bileşen tanımlamaları daha basit ve daha az kod yazımını gerektirmektedir. Vue, template yapısını kullanarak HTML benzeri bir yapı sunmakta, bu da bileşenlerin daha kolay okunmasını ve yönetilmesini sağlamaktadır.
- *Tek Dosya Bileşenleri (Single File Components):* HTML, JavaScript ve CSS'in tek bir dosyada birlikte kullanılmasını destekleyen tek dosya bileşenlerini (Single-File Component, SFC) bir arada sunmaktadır. Bu yaklaşım, bileşenleri izole etmekte, kodun modüler ve yönetilebilir olmasını sağlamaktadır. React'ta, benzer bir yapı JSX kullanılarak elde edilebilse de Vue.js'in SFC yaklaşımı daha entegre ve düzenli olarak değerlendirilmektedir.
- *Topluluk ve Eklenti Desteği:* Vue.js, geniş bir topluluğa ve birçok üçüncü parti eklentiye sahiptir. Bu eklentiler, geliştirme sürecini hızlandırmakta ve çeşitli projeler için hazır çözümler sunmaktadır. Özellikle Vue Router ve Vuex gibi resmi eklentiler, uygulama geliştirmeyi daha da kolaylaştırmaktadır.

Bu özelliklerinden faydalanarak ARKHESTOR yazılımı geliştirilirken, Vue.js'de yer alan Pinia, QR Kod okuyucu, QR Kod oluşturucu, Google haritalar gibi kütüphaneler ve Tailwind CSS kullanılmış, bu sayede görselleştirmeler daha kısa sürede kodlanmış ve oluşturulabilmiştir.

3.1.4 JSON Web Token (JWT)

JSON Web Token'lar (JWT), modern kimlik doğrulama mekanizmaları içerisinde erişim kontrol kimlik bilgilerini güvenli bir şekilde yönetmek ve taraflar arasında aktarmak için yaygın olarak kullanılmaktadır (Niewolski vd., 2021)

Token) kullanarak kimlik doğrulama işlemini gerçekleştirmektedir. Kullanıcı, Vue.js tabanlı frontend'de giriş bilgilerini girmekte ve bu bilgiler Axios aracılığıyla API'ye gönderilmektedir. Laravel, gelen isteği doğrulamakta ve başarılı bir doğrulama durumunda kullanıcıya bir JWT dönmektedir. Bu token, Vue.js uygulamasında Pinia tarafından yönetilen merkezi bir durum (state) deposunda saklanmaktadır. Her korumalı API isteğinde, bu token istemci tarafından otomatik olarak eklenmekte ve sunucu, token'ı doğrulayarak isteğin geçerliliğini kontrol etmektedir. Bu şekilde, kullanıcı her işlem yaptığıında kimliğini yeniden doğrulamak zorunda kalmadan, güvenli bir şekilde uygulamanın korumalı alanlarına erişebilmektedir. Bu yöntem, oturum yönetimini kolaylaştırırken, uygulamanın güvenliğini de üst seviyeye taşımaktadır. Kimlik doğrulama sırasından gerçekleştirilen adımlar, yazar tarafından oluşturulan akış diyagramında (Şekil 3.2) sunulmuştur.



Şekil 3.2. Kimlik doğrulama akış diyagramı

Yukarıda verilen kimlik doğrulama akışından faydalanarak, ARKHESTOR yazılımı geliştirilirken izlenen adımlar sırasıyla şu şekildedir:

Öncelikle, Vue.js ve Pinia state yönetim kütüphanesi kullanılarak kullanıcı, kimlik bilgilerini (kullanıcı adı ve şifre) giriş formuna girmektedir. Bu bilgiler, API'ye gönderilmekte, API, bilgilerin doğru olup olmadığını kontrol etmekte ve başarılı bir giriş yapıldığında, kullanıcı bilgileri ve token'ı tekrar kullanıcıya dönmektedir. Ardından,

dönen kullanıcı bilgileri ve token, Pinia store'da saklanmakta ve localStorage veya sessionStorage gibi tarayıcı depolama yöntemleriyle tarayıcıya kaydedilmektedir. Bu sayede sayfa yenilendiğinde bile kullanıcı oturumu korunmakta; uygulamanın belirli bölümleri veya işlevleri, yalnızca kimliği doğrulanmış kullanıcılar tarafından erişilebilir hale getirilmektedir. Örneğin, bir kullanıcı oturumu açmadan belirli sayfalara erişememekte veya belirli işlemleri gerçekleştirememektedir. Kullanıcı oturumu açtıktan sonra, tüm API isteklerinde token kullanılarak yetkilendirme yapılmaktadır. Böylece, kullanıcı güvenliği ve veri bütünlüğü sağlanabilmektedir.

3.1.6 QR Kod

QR kod sistemleri, ilk olarak 1994 yılında Denso Wave tarafından otomobil parçalarının üretim sürecini takip etmek amacıyla geliştirilmiştir (Masih, 2022). QR kodlar, geleneksel bir barkoddan daha fazla veri depolayabilen, hızlı okunabilir ve esnek bir veri depolama yöntemi sunmaktadır. QR kodlar, metin, URL, iletişim bilgileri, coğrafi konumlar gibi çeşitli veri türlerini saklayabilmektedir. QR kodların oluşturulmasında, kare şeklindeki modüllerin (küçük kare hücreler) düzenlenmesi kullanılmaktadır. Bu modüller, QR kodun veri kapasitesine ve hata düzeltme seviyesine bağlı olarak değişen boyutlarda bir ızgara oluşturmaktadır.

Oluşturulan QR kodun yapısal bileşenleri şunlardır (De La Torre vd., 2015; Tiwari, 2016):

- *Konum Belirleyici İşaretler:* Kodun köşelerinde yer alan ve tarayıcının kodun yönünü ve yerini belirlemesine yardımcı olan üç büyük kareyi temsil etmektedir.
- *Hizalama İşaretleri:* Büyük QR kodlarda, kodun okunmasını sağlamak için konum belirleyici işaretler dışında yer alan daha küçük işaretlerdir.
- *Zamanlama Desenleri:* Kodun satır ve sütunlarını hizalayan ve tarayıcının veri hücrelerini doğru bir şekilde tanımlamasına yardımcı olan düzenlemelerdir.
- *Veri ve Hata Düzeltme Kodları:* QR kodun ana veri kısmını ve hata düzeltme bilgilerini içeren alanlardır. Hata düzeltme, QR kodun bir kısmının hasar görmesi durumunda bile verilerin doğru bir şekilde okunmasını sağlamaktadır.

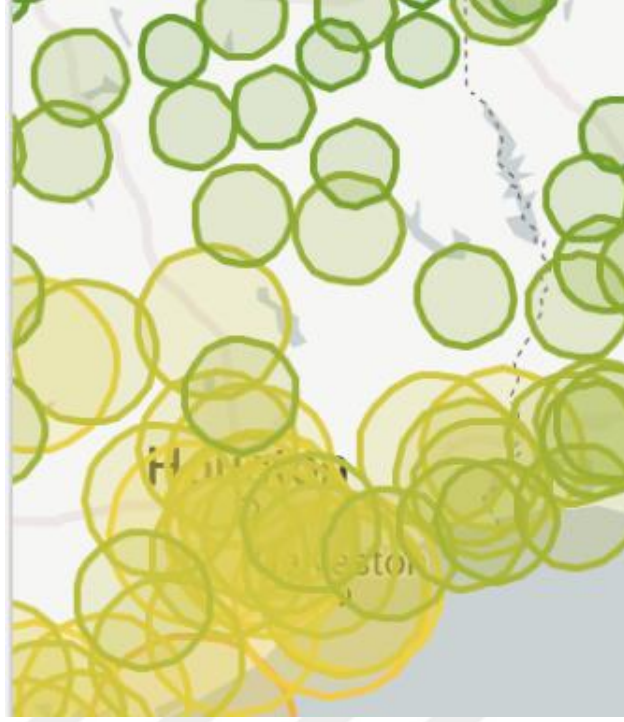
Yüksek veri kapasitesi, hata düzeltme yetenekleri ve hızlı tarama özellikleri ile modern veri depolama ve iletilme yöntemleri arasında önemli bir yer tutan QR kodlar, geliştirilen ARKHESTOR uygulamanın her alanında kullanılmıştır. Örneğin; bir raf, bir kazı alanı, bir buluntu, bir birim için QR Kod oluşturulabilmektedir. Bu sayede geliştirilen uygulama içinden herhangi bir rafa ait QR Kod tarandığında o raftaki tüm

buluntular listelenmekte ve kolayca aranılan buluntuya erişilebilmektedir. Benzer şekilde, bir kazı alanındaki QR Kod taratıldığında o güne kadar oradan çıkarılan tüm buluntular listelenmektedir. Bir buluntu üzerindeki QR Kod okutulduğunda, o buluntuya ait bilgiler, görseller, yapılan restorasyon çalışmalarına doğrudan erişilebilmektedir. Bu da arkeolojik çalışmalarda kazı alanında olursa dahi, uygulamanın bulunduğu ve internet bağlantısı olan bir cihaz aracılığıyla tüm bilgilere anında erişimi sağlamaktadır. Ayrıca geliştirilen uygulama aracılığıyla, tekrar incelenmek istenilen bir buluntunun depoda hangi rafta, hangi kasada, hangi torbada olduğu da kolayca öğrenilebilmektedir. Örneğin, bir arkeolojik araştırmacı geleneksel yöntemlerle kayıt altına alınmış bir buluntuya ulaşmak istediğinde, kayıtlara erişse dahi raflar arasından, kasaları ve torbaları tek tek inceleyerek bu buluntuya ulaşabilmektedir. Ancak ARKHESTOR kullanıldığında, araştırmacı seçtiği buluntunun doğrudan hangi kasada, hangi rafta ve hangi torbada olduğunu kolayca öğrenebilecek ve buluntuya kolayca erişebilecektir.

3.1.7 Harita Servisleri ile Veri Görselleştirme

Web tabanlı bir harita hizmeti uygulamasının altyapısı genellikle HTML5, CSS3 ve JavaScript gibi teknolojileri, Yandex, Google Maps API V3 ve Google Fusion Tables API'sini içermektedir (Orhan vd., 2017). Bu teknolojiler, haritaların web uygulamalarına sorunsuz bir şekilde entegrasyonunu sağlayarak verilerin mekânsal bir bağlamda görselleştirilmesini mümkün kılmaktadır. Ayrıca, Google Maps API, özel veri noktalarının eklenmesini destekleyerek geliştiricilerin harita içeriğini belirli ihtiyaçlara göre özelleştirmelerine olanak tanımaktadır (Yousif ve Ibrahim, 2023).

Geliştirilen ARKHESTOR uygulamasında ise Google Maps API kullanımı tercih edilmiştir. Bu sayede kazı alanları, depolar, buluntuların çıkarıldığı koordinatlar geliştirilen uygulama üzerinde işaretlenmekte ve gösterilmektedir. Böylece arkeoloji alanı çalışanları harita üzerinden kolayca buluntuların takibini yapabilmekte bununla birlikte kazı esnasında çıkartılan bir bulgunun lokasyonu eş zamanlı olarak uygulama üzerinden işaretlenebilmektedir. Ayrıca hangi bölgede hangi buluntuların olduğunun dağılımı, hangi tür buluntuların hangi bölgelerde yoğunlaştığı gibi veriler harita üzerinde görselleştirilerek araştırmacının daha iyi yorumlar yapabilmesine olanak tanımaktadır.



Şekil 3.3. Buluntu lokasyonların Google Maps üzerindeki görünümü

3.2 Backend

3.2.1 PHP

PHP (Hypertext Preprocessor), başlangıçta kişisel bir web sayfası oluşturmak için tasarlanan ve zaman içinde dinamik web geliştirme için güçlü bir araç haline gelen sunucu taraflı bir betik dilidir. PHP, geniş çapta kullanım alanı bulmuş ve özellikle web geliştirme topluluğunda büyük bir popülerlik kazanmıştır. Açık kaynak kodlu bir dil olan PHP, dinamik içerik oluşturma, veri tabanı entegrasyonu ve nesne yönelimli programlama (Object-oriented programming-OOP) desteği gibi özelliklerle öne çıkmaktadır. Güçlü kütüphaneleri ve çerçeveleri ile web uygulamaları geliştirme süreçlerini kolaylaştırmayı amaçlayan PHP için Siame ve Kunda (2017) açık kaynak topluluğunda ve endüstride büyük web tabanlı uygulamalar ve çerçeveler oluşturmada kullanılan geniş çapta tercih edilen bir programlama dili olduğunu belirtmiştir (Siame ve Kunda, 2017). Ayrıca, PHP zaman içinde bir betik dilinden OOP diline evrilmiş, bu da web geliştirme topluluğuna OOP'nin güçlü taraflarının da entegre edilmesini sağlamıştır (Obilikwu ve Terwase, 2021).

PHP'nin kaynak tüketiminin düşük olmasının yanında, sağladığı en önemli avantajların başında bütün platformlarda çalışabilmesi, bilinen tüm veritabanı sistemleriyle entegre olabilmesi, geniş ve ucuz hosting desteği sunması gelmektedir. Düşük kaynak tüketimi,

PHP'nin sunucular üzerinde verimli çalışmasını sağlamakta ve yüksek trafikli web sitelerinde bile performans kaybı yaşanmasını önlemektedir. Ayrıca PHP'nin bellek yönetimi ve optimize edilmiş kod yapısı, sunucu kaynaklarının etkin kullanımını sağlamaktadır. Bu özellikler, PHP'yi büyük ölçekli web uygulamaları ve yoğun veri işleme gerektiren projeler için ideal bir seçim haline getirmektedir. Sağladığı avantajlar sebebiyle geliştirilen ARKHESTOR yazılımı için PHP ile geliştirilmiş bir framework ile çalışmaya karar verilmiştir.

3.2.2 Framework

Framework, yazılım geliştirme süreçlerinde kullanılan bir yapıdır ve belirli bir programlama dili veya teknoloji seti üzerine inşa edilmiştir. Framework'ler, yazılım geliştirme süreçlerini hızlandırabilmekte, kod tekrarını azaltabilmekte, kodlama standartlarını oluşturabilmektedir. Genel olarak değerlendirildiğinde, geliştirme sürecinin daha verimli yürütülmesine olanak sağlamaktadır.

PHP dilinde framework kullanımı yaygın olarak tercih edilmektedir. PHP dilindeki framework'ler, geliştiricilere hazır çözümler sunarak tekrar eden işleri azaltmakta ve projelerin daha organize bir şekilde geliştirilmesine olanak tanımaktadır. Örneğin, Laravel, Symfony, CakePHP, Yii ve CodeIgniter gibi popüler PHP frameworkleri, geliştiricilere MVC (Model-View-Controller) mimarisini kullanarak projelerini kolayca yapılandırma ve yönetme imkânı sağlamaktadır (Niarman vd., 2023). Ayrıca, framework kullanmanın avantajlarından bir diğeri, güvenlik ve performans gibi konularda standartların oluşturulmasına imkân tanınmasıdır. Framework'lerin genellikle güvenlik önlemlerini, en iyi ve en yeni kütüphaneleri içermesi sebebiyle frameworkler geliştiricilerin güvenlik açıklarını azaltmalarına ve daha güvenli uygulamalar oluşturmalarına yardımcı olmaktadır. Bunun yanında framework'ler genellikle performans optimizasyonları sağlamakta ve kodun daha verimli çalışmasına olanak tanımaktadır (Yang ve Aklani, 2023).

Sonuç olarak, framework kullanımı yazılım geliştirme süreçlerini düzenlemekte, hızlandırmakta ve kodlama standartlarını sağlamaktadır. Framework'ler, geliştiricilere hazır çözümler sunarak projelerin daha etkili bir şekilde geliştirilmesine olanak tanımakta ve genel olarak yazılım geliştirme süreçlerini iyileştirmektedir. Bu noktadan hareketle ARKHESTOR yazılımı geliştirilirken framework kullanılmasına karar verilmiştir.

3.2.3 Laravel

Laravel Framework'ü Model-View-Controller yapısına sahip, kimlik doğrulama, yönlendirme, oturum yönetimi, önbellekleme, api servisleri ve diğer bileşen işlevleri dahil olmak üzere web geliştirme sürecinin çeşitli yönlerini basitleştirmekte, yazılan kodun anlaşılmasını ve çalışılmasını kolaylaştırmaktadır (Sukma vd., 2023).

Ayrıca Laravel framework'ünün daha hızlı performans, migration ara katmanı aracılığıyla SQL erişimine ihtiyaç duymadan veri tabanı yapısının sürüm kontrolünü sağlamak(Sendiang vd., 2018), gelişmiş veri güvenliği ve HMVC (Hierarchical Model View Controller) konseptiyle Blade şablonlama gibi gelişmiş özelliklerin kullanımı geliştiriciye avantajlar sunmaktadır (Alamsyah vd., 2024).

Laravel'in temel güçlü yönlerin arasında, databasede tablo şemalarının oluşturulmasını kolaylaştıran migration ara katmanı özelliğine sahip olmasıdır. Migration ara katmanı sayesinde database oluşturma süreci, database işlemlerinin geri alınması, database yapısının güncellenmesi süreci hızlandırılmakta ve daha takip edilebilir hale getirilmektedir (Alamsyah vd., 2024). Ayrıca, Laravel, geliştiricilerin uygulamalarında çeşitli işlevleri verimli bir şekilde uygulamaları için kullanıma hazır kütüphanelerle güvenilir bir ekosistem sağlamaktadır (Alamsyah, 2024). Frameworkün belgeleri, şifreleme paketleri, birim testi desteği ve geliştirme ekosistemi, popüleritesine ve kullanılabilirliğine daha fazla katkıda bulunmaktadır (Ananda vd., 2023).

AREKSTOR uygulaması geliştirilirken backend olarak Laravel'i seçmemin birkaç nedeni bulunmaktadır:

- *Performans ve Güvenlik:* Laravel, hızlı performans ve kararlı veri yeniden yükleme özellikleri ile büyük miktarda arkeolojik verinin etkin bir şekilde işlenmesini sağlamaktadır. Ayrıca, gelişmiş veri güvenliği ile hassas arkeolojik verilerin korunmasını, uygulamanın güvenli ve verimli çalışmasını kolaylaştırmaktadır.
- *Database Migration Yönetimi:* Laravel'in veritabanı versiyon yönetimi özelliği, veritabanı kurulum sürecini hızlandırarak veritabanı işlemlerinin geri alınması ve yapısının güncellenmesini kolaylaştırmaktadır. Bu özellik sayesinde, arkeolojik veriler düzenli bir şekilde saklanabilmekte ve gerektiğinde kolayca güncellenebilmektedir.
- *Modüler Yapı ve Esneklik:* HMVC (Hierarchical Model View Controller) konseptiyle birlikte Blade şablonlama, uygulamanın esnek ve modüler bir yapıya

sahip olmasını desteklemekte; böylece uygulamanın geliştirilmesi ve kontrolü daha kolay ve yönetilebilir olmaktadır.

- *Geliştirici Ekosistemi:* Laravel'in sağlam ekosistemi, kullanıma hazır kütüphaneler ve kapsamlı dokümantasyon ile çeşitli işlevlerin verimli bir şekilde uygulamaya entegre edilmesini sağlamaktadır. Bu ekosistem yapısı, geliştirme sürecinde zaman tasarrufu sağlarken, aynı zamanda uygulamanın işlevselliğini artırmaktadır.
- *Test Desteği:* Laravel, geliştiricilere birim testi desteği ve şifreleme paketleri sunmaktadır. Bu sayede, uygulamanın sürdürülebilirliği ve güvenilirliği artırılmakta, uzun vadede daha güvenli ve stabil bir çözüm yolu sunmaktadır.
- *API Servisi ve Routing:* Laravel'in API yönlendirme özelliği, RESTful API'lerin oluşturulmasını ve yönetilmesini kolaylaştırmaktadır. Route tanımlamaları sade ve anlaşılır olup, karmaşık API yapılarını bile düzenli ve takip edilebilir kılmaktadır. Bu özellik, arkeolojik verilerin farklı cihaz ve platformlardan erişilebilir olmasını sağlarken, diğer taraftan veriler üzerinde çeşitli işlemler yapılabilmesini kolaylaştırmaktadır.
- *Ön Bellekleme (Caching):* Laravel, farklı önbellek sürücülerıyla çalışabilmekte, bu sayede veri sorgularının önbelleğe alınarak performansın artırılmasını sağlamaktadır. Özellikle sık kullanılan verilerin tekrar tekrar işlenmesinin önüne geçilmesi, arkeolojik kazı alanları uygulamasının daha hızlı ve verimli çalışmasına yardımcı olmaktadır.
- *Kimlik Doğrulama Sistemi (Authentication System):* Laravel, kullanıma hazır kimlik doğrulama sistemleri ile kullanıcı kayıtları, oturum açma ve yetkilendirme işlemlerini kolayca entegre etmektedir. Kullanıcıların güvenli bir şekilde sisteme erişimini ve farklı kullanıcı rollerine göre yetkilendirme işlemlerinin yapılmasını sağlamaktadır. Arkeolojik kazı alanları uygulaması için farklı kullanıcı gruplarının (örneğin yöneticiler, araştırmacılar ve ziyaretçiler) farklı erişim haklarına sahip olması gibi tanımlama ihtiyaçlarını Laravel'in kimlik doğrulama sistemi karşılamaktadır.
- *Model Yapısı:* Laravel'in Eloquent ORM (Object-Relational Mapping) sistemi, veritabanı işlemlerini daha kolay ve anlaşılır hale getirmektedir. Eloquent, her tabloyu ve ilişkili veritabanı kayıtlarını temsil eden modeller kullanmaktadır. Bu modeller, veritabanı kayıtları üzerinde CRUD (Create, Read, Update, Delete)

işlemlerinin kolayca yapılmasını sağlamaktadır. Arkeolojik kazı alanları uygulamasında kazı alanları, eserler, araştırmacılar gibi çeşitli veritabanı tabloları arasında ilişkiler kurulması gerekebildiği için Eloquent ORM, bu ilişkileri tanımlamak ve yönetmek için güçlü bir alt yapı sunmaktadır. Örneğin, bir kazı alanının birden fazla esere sahip olabileceği bir “birden çoğa” ilişkisi veya bir eserin birden fazla araştırmacı tarafından incelenebileceği “çoğundan çoğa” ilişkisi Eloquent ile kolayca yönetilebilmektedir.

Yukarıda bahsi geçen sebepler temelinde Laravel, arkeolojik kazı alanları uygulamasının backend’i için ideal bir seçim olarak ortaya çıkmaktadır. Sağladığı avantajlar, projenin başarısını ve verimliliğini artırarak, arkeolojik verilerin güvenli, düzenli ve hızlı bir şekilde yönetilmesine katkıda bulunmaktadır. Laravel’in API yönlendirme, önbellekleme, kimlik doğrulama sistemi ve model yapısı özellikleri, uygulamanın performansının, güvenliğinin ve esnekliğinin daha da güçlendirilmesini sağlamaktadır.

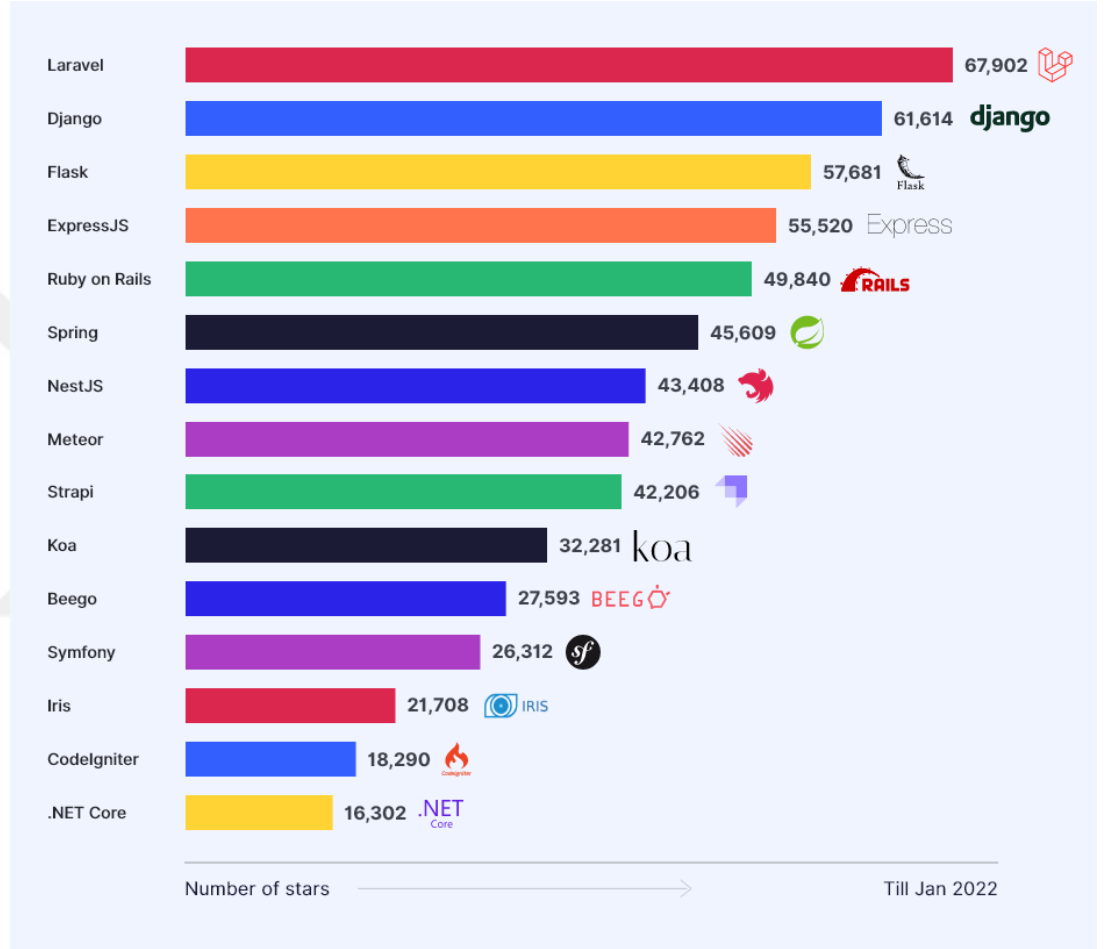
Laravel gibi PHP programlama dilinde geliştirilen ve geliştiriciler tarafından sıklıkla tercih edilen diğer PHP frameworklerinin güçlü ve zayıf yönleri Çizelge 3.1’de sunulmuştur.

Çizelge 3.1. PHP Frameworklerinin güçlü ve zayıf yönleri

Framework	Artıları	Eksileri
Laravel	- Kimlik doğrulama sistemleri	- Daha küçük projeler için özelliklerine ihtiyaç duyulmaması
	- Migration yönetimi	- Öğreniminin zor veya zaman alıcı olması
	- Yüksek performans ve veri güvenliği	
	- Modüler yapı ve Blade şablonlama	
	-Kapsamlı ekosistem ve dokümantasyon	
Symfony	-Esneklik ve yeniden kullanılabilir bileşenler	- Başlangıçta performans optimizasyonu gerekliliğinin olması
	-Kapsamlı ve yapılandırılmış dokümantasyon	- Öğreniminin zor veya zaman alıcı olması
	- Aktif ve destekleyici topluluk	
	- Gelişmiş güvenlik özellikleri	
Yii	-Yüksek performans ve hızlı geliştirme	- Küçük geliştirici topluluğu
	-Kod jeneratörleri ile hızlı prototipleme	- Daha az kapsamlı dokümantasyon
	- Güçlü güvenlik bileşenleri	
Zend	- Modüler yapı	- Öğreniminin zor veya zaman alıcı olması
	- Kurumsal düzeyde destek	- Küçük geliştirici topluluğu

- Yüksek yapılandırılabilirlik	- Kompleks yapı
Not. Tablo (Stauffer, 2023), (Abutaleb vd., 2021), (Bagwan ve Ghule, 2019), (Greiff ve Johansson, 2019), (Norhaidah, 2019)ve (Porebski vd., 2011) 'den faydalanarak yazar tarafından oluşturulmuştur.	

Bununla birlikte, 2022 yılı için <https://webmobtech.com/blog/why-choose-laravel-as-the-best-php-framework/> web sitesinde Laravel'in, GitHub'da aldığı yıldızlara göre yapılan sıralamada birinci sırada olduğu görülmektedir.



Şekil 3.4. Framework kullanıcı deneyim puanları

3.2.4 Veritabanı: MySQL

MySQL, dünya çapında yaygın olarak kullanılan açık kaynaklı bir ilişkisel database yönetim sistemidir (RDBMS). 1995 yılında MySQL, AB tarafından geliştirilmiş olup, şu anda Oracle Corporation tarafından desteklenmektedir. MySQL, SQL (Structured Query Language) dilini kullanarak veri yönetimi sağlamak ve database oluşturma, düzenleme ve yönetme işlemlerini gerçekleştirmektedir. Şu şer, MySQL databaseinin performansını optimize edebilmek için sorgu optimizasyonu, sunucu

parametre ayarları, veri normalizasyonu, önbellekleme ve replikasyon gibi yöntemlerin kullanılabileceğini belirtmektedir (Šušter ve Ranisavljević, 2023).

MySQL'in geniş kullanıcı tabanı, esnek veri depolama yapısı ve performans optimizasyonu imkanları, diğer database sistemlerine göre tercih edilmesini sağlayan önemli avantajları olduğu görülmektedir. MySQL, birçok farklı sektörlerde, özellikle biyoinformatik, e-ticaret, finans ve sağlık gibi alanlarda yaygın olarak kullanılan güvenilir bir database sistemi olmasından ötürü geliştirilen yazılımda verilerin saklanması için tercih edilmiştir.

3.2.5 OpenCV

OpenCV, görüntülerin okunması, işlenmesi ve ön işleme adımlarında yaygın olarak kullanılan bir kütüphanedir. ARKHESTOR yazılımı geliştirme sürecinde, OpenCV kullanılarak görüntülerin renk dönüşümü, boyutlandırma ve dizi formatına dönüştürülmesi gibi işlemler gerçekleştirilmiştir. OpenCV, görüntü ve video okuma, yazma, yeniden boyutlandırma, renk dönüşümü gibi işlemleri kolayca gerçekleştirmek için kullanılmıştır (Ibañez vd., 2023). Özellikle, görüntülerin farklı renk uzaylarına dönüştürülmesi, boyutlandırılması ve işlenmesi gibi adımlar, OpenCV'nin sağladığı işlevlerle etkili bir şekilde gerçekleştirilebilmektedir (Alhakbani vd., 2023). Bu sayede, görüntülerin işlenmesi ve analiz edilmesi süreçleri daha verimli ve etkili bir şekilde yürütülebilmektedir.

Bu özellikler göz önüne alındığında, OpenCV, görüntülerin işlenmesi ve analiz edilmesi süreçlerinde kullanılan güçlü bir araçtır. Renk dönüşümü, boyutlandırma ve diğer ön işleme adımlarında OpenCV'nin sağladığı işlevler, görüntü işleme projelerinde verimliliği artırması ve işlemleri kolaylaştırma özelliklerinden dolayı yazılımda kullanılan benzer imajları bulma ekranı için imajları yeniden boyutlandırma gibi işlemler için tercih edilmiştir.

3.2.6 TensorFlow ve EfficientNetB0

EfficientNetB0 modeli, görüntülerden özellik çıkarmak için kullanılmış olup, bu süreçteki etkinliği ve verimliliği ile öne çıkmaktadır (Baydogan vd., 2023). Ayrıca, gelişmiş mimarisi ve yetenekleri sayesinde özellik çıkarım sürecini basitleştirilmesini sağlamaktadır (Baydogan vd., 2023). TensorFlow ise modelin yüklenmesi, eğitilmesi ve tahminlemenin yapılması gibi görevlerde kullanılarak, modelle ilgili işlemleri etkin bir şekilde yönetmek için sağlam bir çerçeve sunmaktadır (Taner vd., 2021). Bununla birlikte

TensorFlow, geniş çapta kullanılan bir derin öğrenme çerçevesi olarak, modelin yüklenmesi, verilerle eğitilmesi ve öğrenilen özelliklere dayalı tahminler yapılması gibi çeşitli işlemleri desteklemektedir (Taner vd., 2021). EfficientNetB0 ve TensorFlow'un birleşimi, görüntü işleme ve derin öğrenme görevlerinde sorunsuz bir iş akışı sağlamaktadır.

EfficientNetB0 modelinin görüntülerden özellik çıkarmada kullanılması ve TensorFlow'un model yönetimi ve tahmin görevlerinde kullanılması, ileri düzey araçların verimli görüntü işleme ve derin öğrenme uygulamaları için etkili bir yaklaşım sergilemesinden ötürü ARKHESTOR yazılımında benzer buluntu imajların eşleştirilmesi aşamasında tercih edilmiştir.

3.2.7 Benzerlik Fonksiyonları ve EfficientNetB0, ResNet50, VGG16, InceptionV3 Ön Eğitimli Modeller ve Benzerlik Skorlarını Hesaplayan Servis

Verilerin benzerlik analizinde, Jaccard, Overlap, Cosine, Pearson ve Euclidean gibi benzerlik fonksiyonları, veri noktaları arasındaki benzerliği ölçme, kümeleme, sınıflandırma ve öneri sistemleri, grafik veri bilimi alanı gibi uygulamalarda yaygın olarak kullanılmaktadır. Bu benzerlik ölçümleri, belirli veri türleri ve analiz ihtiyaçlarına yönelik olarak geliştirilmekte ve veri kümeleri içindeki ilişkiler ve desenler hakkında önemli bilgiler sağlamaktadır. Her bir fonksiyon belirli veri türleri ve analiz gereksinimleri için uygundur. Örneğin Jaccard, iş birliği için paydaşları eşleştiren bir modelde benzerlik indeksi için kullanılmaktadır (Kabasakal ve Soyuer, 2021) ve benzerlik ölçümlerinin desen tanıma, makine öğrenmesi ve bilgi erişiminde uygulanabileceği belirtilmektedir (Gardner vd., 2014).

Benzerlik ölçümlerinin uygulama alanlarından biri öneri sistemleridir. Örneğin, Jaccard Similarity ve Cosine Similarity algoritmalarının, içerik tabanlı filtreleme yöntemlerinde kullanılarak öneri sistemlerinde etkili olduğu gösterilmiştir (Sukestiyarno vd., 2023). Ayrıca, Jaccard benzerliğinin, diğer temel algoritmalara göre daha yüksek doğruluk ve sağlamlık sağlayarak görüntü analizi gibi alanlarda kullanıldığı bildirilmiştir (Jiang vd., 2022).

Benzerlik ölçümlerinin kullanım alanları yalnızca öneri sistemleriyle sınırlı değildir. Örneğin, Jaccard benzerliğinin, çeşitli endüstrilerde iş yüklerini hızlandırma konusunda önemli etkileri gözlenmiştir (Sachdeva vd., 2009).

Bunun yanı sıra, cosine benzerliği, görüntü tabanlı içerik erişim sistemlerinde, nesne tespiti, belirli görüntü modüllerine en yüksek benzerliğe sahip optimal bölgeyi

bulmak kullanılmış ve görüntü özelliklerini karşılaştırmada ve benzer görüntüleri getirmede verimli olduğu gözlenmiştir (Aguilar vd., 2021; Zou vd., 2017). Ayrıca, yüz tanıma alanında, cosine benzerliği eşleştirme kriteri olarak kullanılmış ve tanıma görevlerinde yüksek başarı gösterdiği gözlemlenmiştir (Hanmandlu vd., 2013).

Cosine benzerliğinin nesneler veya özellikler arasındaki benzerliği ölçmedeki çok yönlülüğü ve etkinliği, çeşitli alanlarda geniş çapta uygulanmasını sağlamış, bu da onu geniş bir araştırma alanında değerli bir araç haline getirmiştir. Bu nedenle, tez kapsamında geliştirilen ARKHESTOR yazılımında benzer görüntülerin çıkarımı için cosine benzerliğinin kullanılmasına karar verilmiştir.

Cosine benzerliği, çok boyutlu bir uzayda iki vektörün benzerliğini belirlemek için kullanılan bir ölçüttür. Bu ölçüt, iki vektör arasındaki açının kosinüsünü hesaplayarak -1 ile 1 arasında bir değer sağlamaktadır. X ve Y vektörlerinin karşılaştırılması için kullanılan cosine benzerliği formülü, matematiksel olarak vektörlerin iç çarpımının normlarının bölümüdür (Singh vd., 2020). Cosine benzerliği formülü aşağıda Denklem 3.1’de verilmiştir.

$$\cos(\mathbf{X}, \mathbf{Y}) = \frac{\mathbf{XY}}{\|\mathbf{X}\| \|\mathbf{Y}\|} = \frac{\sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i}{\sqrt{\sum_{i=1}^n (\mathbf{x}_i)^2} \sqrt{\sum_{i=1}^n (\mathbf{y}_i)^2}} \quad (3.1)$$

Derin öğrenme ve görüntü işleme bağlamında, kosinüs benzerliği, görüntülerden çıkarılan özellik vektörleri karşılaştırmak için kullanılabilir. Amin’in de çalışmasında belirttiği gibi görüntüler, konvolüsyonel sinir ağları (CNN’ler) veya EfficientNetB0, ResNet50, VGG16 ve InceptionV3 gibi önceden eğitilmiş modeller aracılığıyla özellik vektörleri olarak temsil edilebilmektedir (Amin vd., 2022). Bu şekilde, kosinüs benzerliği, çıkarılan özellik vektörlerine dayalı olarak görüntüler arasındaki benzerliği niceliksel olarak belirlemeye yardımcı olabilmektedir. TensorFlow, bu özellik vektörlerini yüklemek, işlemek ve karşılaştırmak için kullanılabilir (Singh vd., 2020).

EfficientNetB0, ResNet50, VGG16, InceptionV3 gibi önceden eğitilmiş modeller, özellik çıkarımı için kullanılırken, TensorFlow model operasyonlarını (yükleme, eğitim ve tahmin görevleri dahil) yönetmek için tercih edilmektedir. Bu araçların gücünden yararlanarak geliştiriciler, görüntü ile ilgili uygulamalarda derin öğrenmeyi etkili bir şekilde işleyebilmektedirler. Kosinüs benzerliği metriği, bu araçlarla birlikte, çok boyutlu bir uzayda görüntü özelliklerini karşılaştırmayı sağlamakta ve görüntü geri çağırma,

benzerlik arama ve içerik tabanlı görüntü analizi gibi görevleri de kolaylaştırmaktadır (Amin vd., 2022).

Bu çalışmada uygulama geliştirme sürecinde, EfficientNetB0, ResNet50, VGG16 ve InceptionV3 modelleri kullanılarak özellik çıkarımı yapılmış; her model ile farklı eşik değerlerinde çeşitli benzerlik fonksiyonları kullanılarak performans değerlendirmesi gerçekleştirilmiştir. Sonuç olarak, EfficientNetB0 modelinin kullanılması karar verilmiştir.

EfficientNetB0 modelinin kullanımına karar verildikten sonra Python programlama dili için mikro bir web frameworkü olan Flask (Grinberg, 2018) kullanılarak ARKHESTOR'un veritabanına eklenen yeni resimlerin belirli aralıklarla kontrol edilip işlenmesini sağlayan bir servis geliştirilmiştir. Bu servis, her 60 saniyede bir çalışarak yeni eklenen resimleri kontrol etmekte ve bu resimleri işleyerek benzerlik skorlarını database'e eklemektedir.

Servisin temel işleyişi şu şekildedir:

- *Son İşlenen ID'nin Kaydedilmesi ve Yüklenmesi:* İşlenen son resmin ID'si yerel bir dosyada saklanmakta ve servis başlatıldığında bu ID yüklenerek en son hangi resmin işlendiği belirlenmektedir. Bu sayede modelin tekrar tekrar sıfırdan eğitilmesine gerek kalmamakta ve performans artışı sağlanmaktadır.
- *Görüntü Ön İşleme:* Servis ilk kez çalıştırıldığında veya yeni resimler eklendiğinde, tüm bu resimler OpenCV kütüphanesi kullanılarak ön işlemden geçirilmektedir. Bu işlem, benzerlik skorlarının hesaplanmasından önce resimlerin boyutlandırılması ve gerekli renk dönüşümlerinin gerçekleştirilmesini sağlamaktadır.
- *Özellik Çıkarımı ve Benzerlik Analizi:* EfficientNetB0 modeli kullanılarak resimlerin özellik vektörleri çıkarılmaktadır. Ardından, bu vektörler kullanılarak cosine benzerliği yöntemi ile veritabanındaki diğer resimlerle benzerlik hesaplamaları yapılmakta ve sonuçlar veritabanına eklenmektedir.
- *Yeni Resimlerin Kontrol Edilmesi:* Her 60 saniyede bir veritabanı kontrol edilerek yeni eklenen resimler belirlenmektedir. Bu resimler işlenerek veritabanına eklenmekte ve benzerlik skorları güncellenmektedir.
- *Güncellemelerin Saklanması:* İşlenen yeni resimlerin ID'leri güncellenerek yerel dosyada saklanmaktadır, böylece bir sonraki kontrol döngüsünde tekrar işlemeye tabi tutulmamaktadırlar.

Geliştirilen benzerlik servisinin backend ile entegrasyonu Flask tabanlı bir API ile sağlanmıştır. Bu sayede API, kullanıcıların yükledikleri resimlerle veritabanındaki mevcut resimler arasında hızlı bir şekilde benzerlik bulmalarını sağlamaktadır. API'ye yüklenen resimler işlenmekte ve benzer resimlere ait buluntuların bilgileri JSON formatında dönüştürülerek ve Vue.js tarafında işlenerek liste halinde kullanıcıya sunulmaktadır.

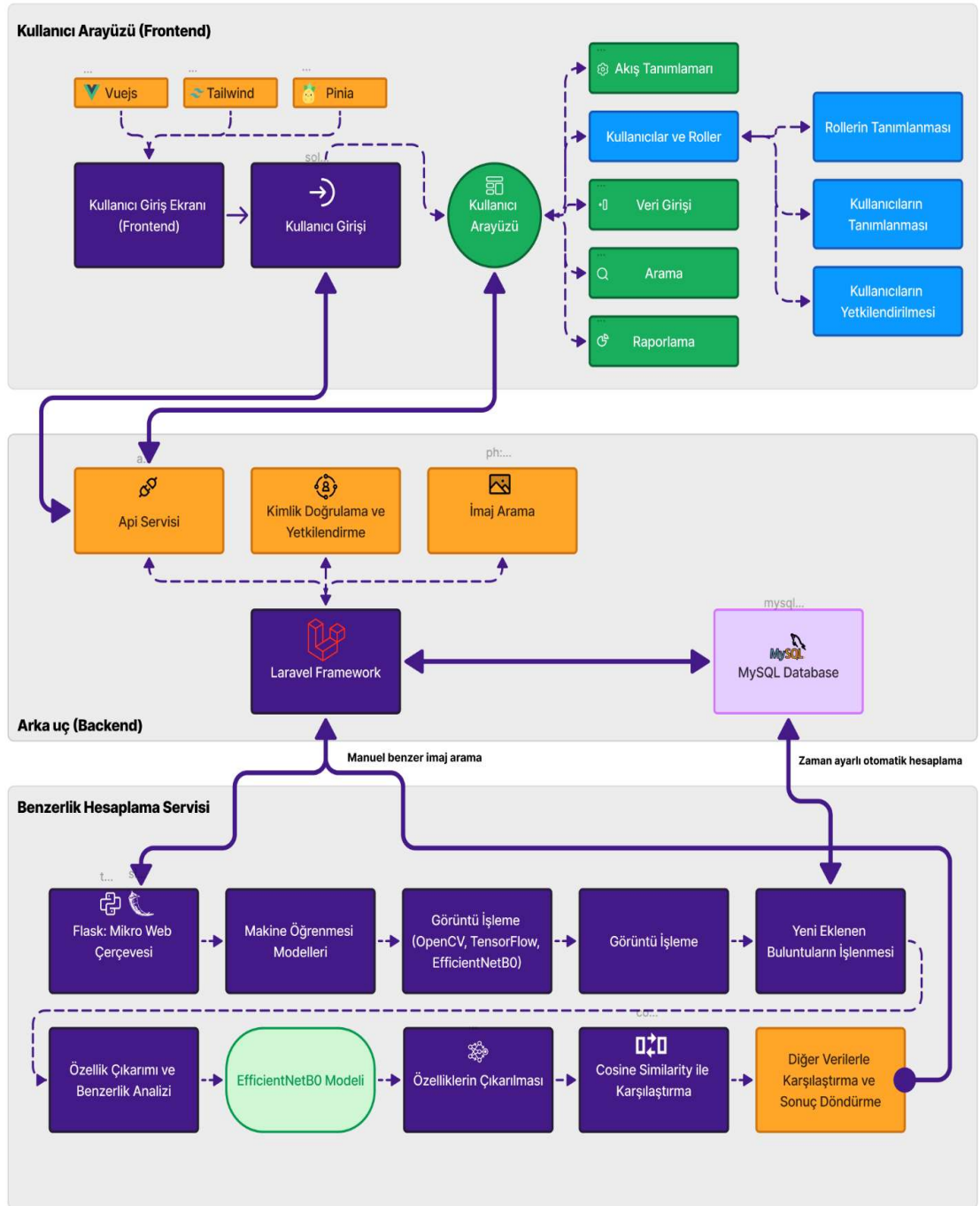
Bu servis sayesinde, yeni eklenen buluntular hızlı ve otomatik bir şekilde işlenerek veritabanının sürekli güncel ve doğru kalması sağlanmaktadır. Ayrıca, arkeologlar ve araştırmacılara, yeni buluntuların mevcut veritabanındaki buluntularla olan ilişkilerinin hızlıca analiz edebilme imkânı sunulmaktadır.

3.2.8 Yayınlama: Nginx

Nginx, düşük bellek tüketimi ve yüksek performans sunan açık kaynaklı bir web sunucu yazılımıdır (Alhakbani vd., 2023). Ters proxy, IPv6, yük dengeleme, FastCGI desteği, web soketleri, statik dosya işleme ve TLS/SSL gibi özelliklere sahiptir (Mardianto vd., 2021). Bu özellikler, Nginx'in yüksek trafikli sitelerde istikrarlı performans göstermesini ve sunucu kesintilerini minimumda tutmasını sağlamaktadır. Nginx, basit yapılandırma dosyaları ve etkili yönetim araçları sayesinde kolayca yapılandırılabilir ve yönetilebilmektedir. Nginx'in düşük donanım gereksinimleri ve yüksek verimliliği, maliyet etkin bir çözüm sunmaktadır. Performansı, hafifliği, ölçeklenebilirliği ve güvenilirliği, Nginx'in tercih edilme nedenleri arasında yer almakta olup bu çalışmada seçim gerekçesini oluşturmaktadır.

3.3 Sistem Mimarisi

Geliştirilen yazılımın mimarisi, arkeolojik buluntuların dijital ortamda kaydedilmesi, sınıflandırılması, analiz edilmesi ve raporlanması amacıyla tasarlanmış entegre bir çözüm sunmaktadır. Bu sistem, kullanıcı arayüzünden başlayarak kazı alanı yapısının tanımlanmasına ilişkin kuralların belirlenmesi, kimlik doğrulama, kullanıcı yetkilendirme, buluntu işlemleri ve benzerlik hesaplama gibi kritik işlevleri içeren çeşitli modüllerden oluşmaktadır. Kullanıcı dostu bir deneyim sağlamak ve envanter kayıtlarının yüksek verimlilik ve performansla saklanmasını temin etmek amacıyla, frontend, backend ve benzerlik servisi için en son teknolojilerden yararlanılmaya özen gösterilmiştir. Şekil 3.5'te geliştirilen yazılımın sistem mimarisi sunulmuştur.



Şekil 3.5. Sistem mimarisi

Şekilde görüldüğü gibi sistemin ön ucu (frontend), kullanıcıların etkileşimde bulunduğu kısımdır ve Vue.js, Tailwind CSS ve Pinia gibi modern web teknolojileri kullanılarak geliştirilmiş bir tek sayfa uygulamasıdır (Single Page Application - SPA). Tailwind CSS, stil bileşenlerinin esnek ve hızlı bir şekilde uygulanması, Pinia, Vue.js için basit ve etkili bir durum yönetimi (state management) sunmaktadır. Pinia yazılımının kullanıcı arayüzünde tutarlı ve hızlı veri güncellemeleri sağlamakta büyük rol

oyunmaktadır. Bu teknolojilerle oluşturulan kullanıcı arayüzü, kullanıcılara kolay ve sezgisel bir deneyim sunarak, veri girişi, arama, raporlama ve kullanıcı yönetimi gibi işlemlerin gerçekleştirmelerine olanak tanımaktadır.

Sistemin arka ucu (backend), veritabanı yönetimi, API servisleri ve iş mantığının yürütüldüğü katmandır. Laravel Framework, bu yapının omurgasını oluşturmakta ve MySQL veritabanı ile entegre çalışarak, verilerin güvenli ve hızlı bir şekilde saklanması sağlamaktadır. Laravel, API servisleri aracılığıyla kullanıcı taleplerini işlemekte, kimlik doğrulama ve yetkilendirme süreçlerini yönetmektedir. Kullanıcıların sisteme erişimlerini güvenli bir şekilde kontrol ederken, aynı zamanda yüksek performanslı veri işlemlerinin sürdürülmesine imkân tanımaktadır.

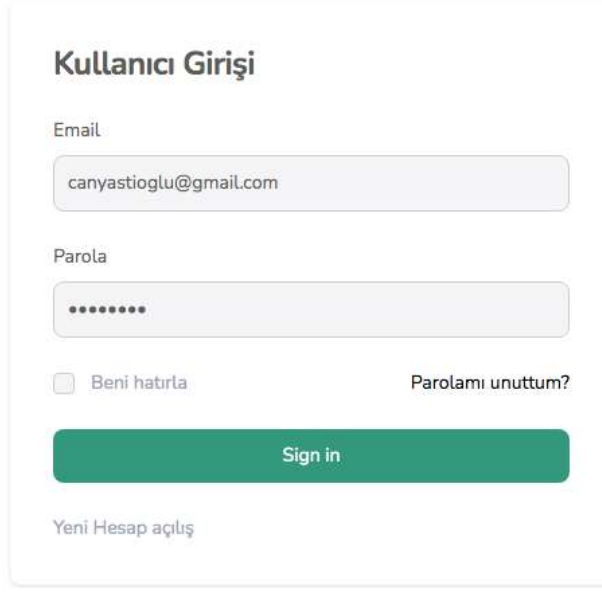
Son olarak ARKHESTOR'un en önemli bileşenlerinden biri, buluntuların görsel özelliklerinin çıkarılması ve benzerlik analizinin yapılmasını sağlayan “Benzerlik Hesaplama Servisi”dir. Bu servis, Python programlama dili ve Flask mikro web çerçevesi kullanılarak geliştirilmiştir. Flask, bu modülün hafif ve hızlı bir şekilde çalışmasını sağlayan esnek bir altyapı sunmaktadır. Görüntü işleme süreçlerinde OpenCV ve TensorFlow gibi güçlü kütüphaneler kullanılırken, EfficientNetB0 modeli ise görsel özelliklerin çıkarılması için tercih edilmiştir. Bu model, yüksek doğruluk oranları ve verimli performansı ile öne çıkmaktadır. Çıkarılan özellikler, Cosine Similarity benzerlik fonksiyonu ile karşılaştırılarak buluntular arasındaki benzerlikler nicel olarak değerlendirilmekte ve veritabanına kayıt edilmektedir. Ayrıca Benzerlik Hesaplama Servisi, belirli aralıklarla (örneğin her 60 saniyede bir) çalışarak veritabanına yeni eklenen buluntuları kontrol etmekte ve bu buluntuları mevcut verilerle karşılaştırarak benzerlik skorlarını hesaplamaktadır. Böylece buluntular arasındaki ilişkilerin hızlı ve doğru bir şekilde belirlenmesini mümkün kıldığı için arkeolojik analizlerde önemli bir avantaj sunması hedeflenmiştir.

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Bu bölümde, ARKHESTOR yazılımının arayüzlerinin, arkeolojik kazı alanlarında verimli bir envanter yönetimi ve kolay kullanım sağlamak üzere nasıl tasarlandığı ele alınmıştır. ARKHESTOR, Vue.js, Tailwind CSS, Pinia, Laravel, Python, Flask gibi modern yazılım teknolojileri ile geliştirilmiş olup, arkeolojik verilerin dijital ortamda güvenli ve verimli bir şekilde saklanması ve yönetilmesi için tasarlanmıştır. Kullanıcıların sisteme erişimden başlayarak, buluntuların kaydedilmesi, sınıflandırılması, analiz edilmesi ve raporlanmasına kadar olan süreçleri destekleyen bu uygulama, arkeologların ve araştırmacıların saha çalışmalarında daha etkili ve hızlı bir şekilde veri yönetimi yapmalarını sağlamaktadır. Uygulamanın sunduğu dinamik kullanıcı arayüzü ve güvenli kimlik doğrulama mekanizmaları, veri güvenliğini en üst düzeyde tutarken, aynı zamanda kullanıcı deneyimini de artırmaktadır. Bu bulgular, ARKHESTOR'un arkeolojik araştırmalarda sadece zaman kazandırmakla kalmayıp, aynı zamanda bilimsel araştırmaların kalitesini ve doğruluğunu da artırdığını göstermektedir. Bununla birlikte sistemin temel özelliklerinden biri olan “Benzerlik Hesaplama Servisi”, arkeologların dijital envanterde yer alan benzer buluntular ile yeni buluntuları hızlı ve güvenilir bir şekilde karşılaştırmalarını mümkün kılmaktadır. Bu da araştırmacılara, sahada daha az zaman harcayarak daha fazla veriye ulaşma imkânı sunmaktadır. Geliştirilen bu sistemin, arkeoloji alanında önemli bir yenilik sunduğu ve sahada çalışan araştırmacılar için büyük bir kolaylık sağladığı değerlendirilmektedir. Buradan hareketle ARKHESTOR envanter takip yazılımının, bu bölüm altında geliştirilme süreci ve elde edilen bulgular aktarılmıştır.

4.1.1 Kullanıcı Girişi (Karşılama)

Kullanıcıların sisteme güvenli bir şekilde erişebilmeleri için, Tailwind CSS ve Vue.js ile dinamik bir kullanıcı arayüzüne sahip bir giriş ekranı tasarlanmış ve uygulanmıştır. Laravel ise güvenli kimlik doğrulama işlevselliğini sağlamak için backend tarafında kullanılmıştır.



Şekil 4.1. Kullanıcı giriş ekranı

Giriş ekranı, Vue.js ve Vue Router kullanılarak tek sayfa uygulama yapısı ile oluşturulmuştur. Bu yapı, sayfa yenilemelerini ortadan kaldırarak ve uygulamanın genel performansını artırarak hızlı ve akıcı bir kullanıcı deneyimi sunmayı hedeflemektedir. Kullanıcı adı ve parola giriş alanları, giriş butonu ve hata mesajlarını içeren form, HTML, Tailwind CSS ve Vue.js bileşenleri kullanılarak oluşturulmuştur.

Kullanıcı giriş ekranındaki durum (state) yönetimi için modern ve esnek bir durum yönetim kütüphanesi olan Pinia kullanılmıştır. Pinia store, giriş durumunu (başarılı veya başarısız) ve kullanıcı bilgilerini yönetmek için yapılandırılmıştır. Pinia'nın esnek yapısı ve düşük bellek kullanımı, uygulamanın performansını artırırken, kodun daha okunabilir ve bakımının daha kolay olmasını sağlamaktadır.

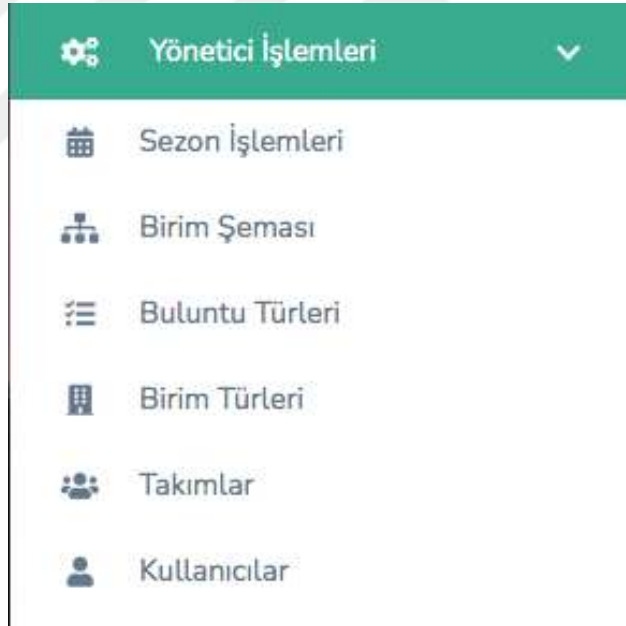
Kullanıcı bilgileri, Laravel'in sunduğu Auth API rotalarına POST isteği ile gönderilmektedir. Laravel Auth yapısı, kullanıcıların kimlik doğrulama işlemlerini yönetmekte, ilgili yönlendirmeleri sağlamakta ve güvenliğin sağlanması için gerekli tüm protokolleri içermektedir. Bu yapı, kullanıcı verilerinin korunmasını temin etmeyi sağlamaktadır.

Routing işlemleri, Vue Router kullanılarak gerçekleştirilmiş ve kullanıcıların oturum açma, oturum kapatma ve kaydolma gibi işlemleri kolaylıkla yönetilmesini sağlamıştır. Vue Router ile sağlanan yönlendirme, kullanıcıların uygulama içinde kesintisiz bir şekilde gezinmelerine olanak tanımış ve kullanıcı deneyimini geliştirmiştir.

Sonuç olarak, bu entegrasyon ile kullanıcıların sisteme güvenli bir şekilde erişmeleri, sistemde kalmaları ve uygulamayı sorunsuz bir şekilde kullanmaları sağlanmıştır. Aynı zamanda geliştiriciler için de esnek ve sürdürülebilir bir kod tabanı sunularak bakım ve güncelleme işlemleri kolaylaştırılmıştır. Bu sayede hem kullanıcı hem de geliştirici açısından önemli faydalar sağlayarak uygulamanın genel performansını ve güvenliğini artırılmıştır.

4.1.2 Tanımlamaların Yapılması

Başarılı kimlik doğrulamasının ardından kullanıcılar, rollerine göre özelleştirilmiş bir yapılandırma arayüzüne yönlendirilmektedir. Arayüzde, sistemin kullanıma başlanması için öncelikle yapılması gereken tanımlamalar bulunmaktadır. İlk giriş yapan kişi kullanıcı rolüne atanacağı için Sezon İşlemleri, Birim Şeması, Buluntu Türleri, Birim Türleri, Takımlar, Kullanıcılar (ve Roller) tanımlamalarının bu kişi tarafından yapılması gerekmektedir.

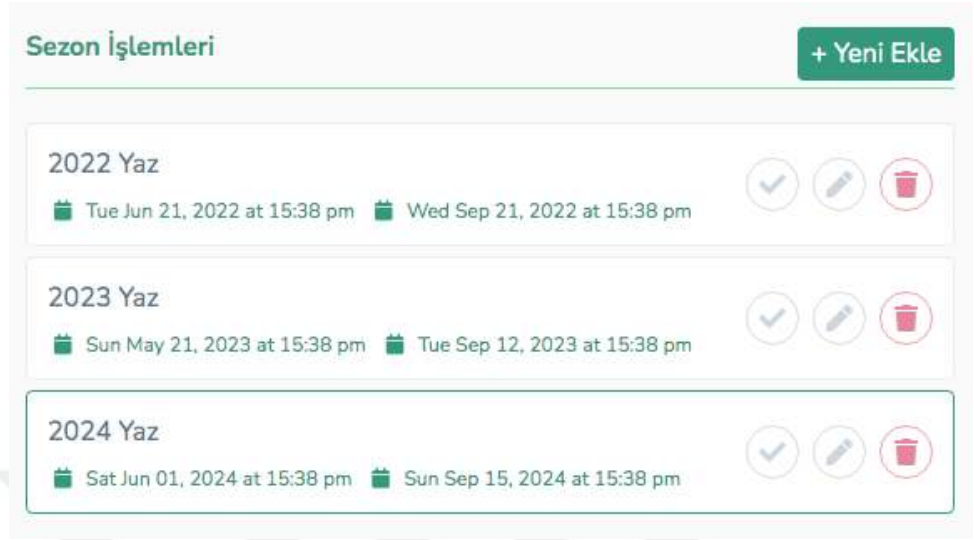


Şekil 4.2. Yönetici işlemleri menüsü

4.1.3 Sezon Tanımlanması

Sezon işlemleri, sistemin zaman ve iş akışını düzenleyen temel yapılandırma adımlarından biridir. Her kazı sezonu için başlangıç ve bitiş tarihleri belirlenir ve ilgili sezon aktif hale getirilir. Bu sayede, kaydedilen her buluntu otomatik olarak aktif sezona atanır. Belirli bir sezona ait buluntuların kolayca filtrelenebilmesi, raporlama ve analiz

süreçlerini kolaylaştırırken veri toplama ve değerlendirme aşamalarında da bütünlük sağlamaktadır.



Şekil 4.3. Kazı sezonlarının tanımlanması

4.1.4 Birim Türlerinin Tanımlanması

Görselde (Şekil 4.4) sunulan ekran, arkeolojik veri yönetim sisteminde kullanılan birimlerin ve bu birimlerin hiyerarşik ilişkilerinin tanımlandığı bir arayüzdür. Bu ekran, kullanıcıların birim türlerini ve bu türlerin altına gelebilecek diğer birim türlerini belirlemelerine olanak tanımaktadır. Birim şeması oluşturulurken, bu tanımlamalar kullanılarak hiyerarşik bir yapı oluşturulacaktır.

Yapı İlişkileri
Yapılar ve alt yapıların doğru bir şekilde oluşturulması için hiyerarşinin tanımlanmasını yaptırdığı bölümdür.

Yapılar	Birim Altına Gelebilecek Yapılar	Eklenebilecek Yapılar
Birim	Laboratuvar	+ Birim
Laboratuvar	Kazı Alanı	+ Kasa
Kazı Alanı	Depo	+ Torba
Depo	Raf	+ Buluntu
Raf	Müze	+ Lokus
Kasa		+ Context
Torba		
Müze		
Buluntu		
Lokus		
Context		

Şekil 4.4. Birim türlerinin tanımlanması ve birim yapılarının oluşturulması

Ekranın sol tarafında, mevcut birim türleri listelenmiştir. Yönetici, bu listeden bir birim türünü seçerek, seçilen birimin altına eklenebilecek birim türlerini orta sütunda tanımlayabilmektedir. Sağ sütunda ise, mevcut birim türleri arasından seçilerek belirlenen birim türüne eklenebilecek yeni birim türleri listelenmiştir.

Bu yapılandırma, sistem yöneticisi veya kullanıcıların birim şemasını oluştururken belirli kurallara göre hareket etmelerini sağlamaktadır. Örneğin, tür olarak ‘Birim’ seçildiğinde, bu birimin altına eklenebilecek türler arasında ‘Depo’, ‘Laboratuvar’, ‘Kazı Alanı’, ‘Raf’ ve ‘Müze’ gibi seçenekler bulunmaktadır. Böylece, kullanıcı birim şemasında yeni bir birim eklemek istediğinde, sadece bu tanımlanmış seçenekler arasından seçim yapabilmektedir.

Birim yapısı, arkeolojik buluntuların ve bunların depolanmasının düzenli ve sistematik bir şekilde yönetilmesine yardımcı olmaktadır. Aynı zamanda, veri girişinde tutarlılığı ve doğruluğu artırarak, araştırmacıların verileri daha verimli bir şekilde organize etmelerini ve analiz etmelerini sağlar. Bu tanımlama süreci, hiyerarşik veri yapılarını destekleyen birimlerin doğru ve tutarlı bir şekilde oluşturulmasına olanak tanımaktadır.

4.1.5 Buluntu Türlerinin Tanımlanması

Bu ekran, buluntu türlerinin tanımlanması, arkeolojik verilerin sistematik bir şekilde organize edilmesine ve analiz edilmesine önemli katkılar sağlamaktadır. Arkeolojik buluntuların sınıflandırılması ve yönetimi için kullanılacak buluntu türlerinin tanımlandığı bir arayüz sunmaktadır. Bu ekran aracılığıyla, sistem yöneticisi farklı buluntu türlerini ve bu türlerin alt kategorilerini oluşturabilmekte, düzenleyebilmekte, tutup sürükleyerek sıralayabilmekte ve silebilmektedir.

Buluntu türlerinin önceden belirlenmesi, bir buluntu eklenirken buluntu türlerinin tanımlanması, veri girişinin standart ve sistematik bir şekilde yapılması imkanını sunmaktadır. Bu, veri tutarlılığını artırmakta ve verilerin daha kolay yönetilmesine olanak tanımaktadır. Tanımlanmış buluntu türleri, kullanıcıların veri girişinde belirli standartlara uymalarını sağlamakta ve bu sayede verilerin kalite ve doğruluğu korunmaktadır. Önceden tanımlanmış buluntu türleri sayesinde, kullanıcılar veri girişi sırasında mevcut kategoriler arasından hızlıca seçim yapabilmektedir. Buluntu türlerinin önceden tanımlanmış olması, veri giriş sürecini hızlandırmakta ve kullanıcıların iş yükünü azaltmaktadır. Aynı zamanda, kullanıcıların önceden tanımlanmış birimler arasından seçim yapabilmesi veri girişinde hata yapma olasılığını da azaltmaktadır. Bunun yanında veri analizinin daha kolay ve etkili bir şekilde yapılmasını; farklı buluntu türleri arasındaki ilişkilerin ve dağılımların, daha net ve anlaşılır bir şekilde analiz edilebilmesini ve grafiksel olarak görselleştirilebilmesini sağlamaktadır. Bu sayede, bilimsel araştırmalarda ve/veya farklı raporlarda kullanılacak verinin kalitesi ve doğruluğu arttırılmaktadır. Örneğin, kullanıcılar ‘Seramik’ ana kategorisi altında ‘Kaplار’, ‘Tabaklar ve Kaseler’, ‘Heykeller’ ve ‘Seramik Parçaları’ gibi alt kategoriler tanımlayabilmektedir. Benzer şekilde, ‘Metaller’ ana kategorisi altında ‘Silahlar’, ‘Aletler’ ve ‘Takılar’ gibi alt kategoriler oluşturulabilmektedir. Bu sayede, arkeolojik buluntuların sınıflandırılması kolaylaşmakta, veriler hiyerarşik bir şekilde organize edilmekte ve uygulama üzerinden raporlaştırılmaktadır.

Sonuç olarak, bu ekran aracılığıyla tanımlanan buluntu türleri, arkeolojik verilerin daha düzenli, tutarlı ve analiz edilebilir bir şekilde yönetilmesine olanak tanımaktadır. Bu yapı aracılığıyla, araştırmacılar ve arkeologlar veri girişini, yönetimini ve analizini daha etkili ve verimli bir şekilde yapabilmektedir.

4.1.6 Kullanıcıların ve Rollerin Tanımlanması

Bu ekran (Şekil 4.5), arkeolojik veri yönetim sisteminde kullanıcıların ve rollerinin tanımlanması ve yönetilmesi için tasarlanmış bir arayüz sunmaktadır. Kullanıcıların sisteme erişimini ve yetkilerini kontrol etmek, güvenli ve düzenli bir veri yönetimi için hayati öneme sahiptir. Bu arayüz, kullanıcıların oluşturulması, düzenlenmesi ve silinmesi ile farklı rollerin atanmasını sağlamaktadır.

Kullanıcılar

Kullanıcıların oluşturulduğu ve yetkilendirildiği bölümdür.

Takımlar

+ Yeni Kullanıcı

 Can YASTIOĞLU Süper Admin  

 Canan YASTIOĞLU Müze  

 Cem KÖK İzleyici  

 Gonca TİRYAKİ Admin  

 Merve AKBULUT  

 Seher YASTIOĞLU  

Kullanıcı Bilgileri



Kullanıcı Yetki/Yetkileri

Admin

İzleyici

Müze

Normal Kullanıcı

Süper Admin

Ad Soyad

Can YASTIOĞLU

Email

canyastioglu@xxxx.com

Cep Telefonu

+90505 111 11|11

Yeni Parola

...

Kaydet

Şekil 4.5. Kullanıcı tanımlama ekranı

4.1.6.1 Kullanıcıların Tanımlanması ve Yönetimi

Şekil 4.5'te yer alan kullanıcı tanımlama ekranının sol bölümünde mevcut kullanıcılar listelenmektedir. Kullanıcıların isimleri, atanmış rollerine göre farklı renklerle veya etiketlerle gösterilmekte, böylece kullanıcılar arasında hızlıca ayırım yapılabilir. 'Yeni Kullanıcı' butonu kullanılarak yeni kullanıcılar tanımlanmaktadır. Mevcut kullanıcılar üzerinde düzenleme yapmak veya silmek için ilgili butonlar kullanılmaktadır.

Kullanıcı ekranının sağ bölümünde ise seçili kullanıcının detaylı bilgileri görüntülenmekte ve düzenlenebilmektedir. Bu bilgiler arasında kullanıcının adı, soyadı, e-posta adresi, telefon numarası ve şifre gibi temel bilgiler yer almaktadır. Kullanıcılar bu bilgilerle birlikte sisteme erişim sağlamak ve ihtiyaç duyulan gerekli değişiklikler bu ekrandan yapılabilir.

4.1.6.2 Roller ve Yetkiler

Kullanıcılara atanan roller, bu ekranın önemli bir parçasını oluşturmaktadır. Rollerin tanımlanması ve kullanıcılarla ilişkilendirilmesi, sistemdeki veri güvenliğini ve iş akışını düzenlemektedir. Kullanıcı rolleri ve yetki seviyeleri şu şekilde tanımlanabilir:

- *Süper Admin*: Sistem üzerinde tam yetkiye sahip olan kullanıcı. Tüm işlemleri gerçekleştirebilir, diğer kullanıcıları ve rolleri yönetebilir.
- *Admin*: Yüksek düzeyde yetkilere sahip kullanıcı. Kullanıcı yönetimi ve veri düzenlemeleri gibi önemli işlemleri yapabilir.
- *İzleyici*: Sistemdeki verilere erişim sağlayabilir ancak veri üzerinde değişiklik yapamaz. Genellikle denetleme ve raporlama amaçlı kullanılır.
- *Müze*: Belirli bir alan veya birimle sınırlı yetkilere sahip kullanıcı. Örneğin, müze ile ilgili verileri yönetebilir.
- *Normal Kullanıcı*: Sistemdeki temel işlemleri gerçekleştirebilen kullanıcı. Verilere erişim sağlayabilir ve sınırlı düzeyde düzenleme yapabilir.

Tasarlanan kullanıcı tanımlama ekranı ve bu ekranda yer alan roller ve yetkiler uygulamada güvenlik, verimlilik, esneklik, şeffaflık ve izlenebilirlik olmak üzere 4 temel avantaj sunmaktadır.

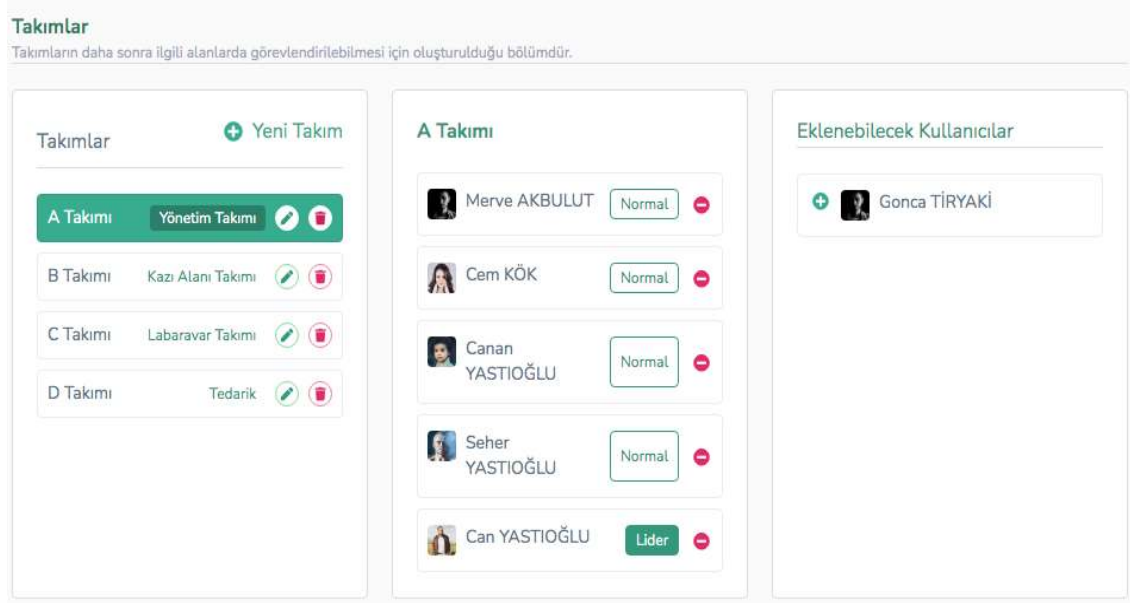
- (1) *Güvenlik*: Kullanıcıların farklı yetki seviyeleri ile atanması, veri güvenliğini sağlamaktadır. Her kullanıcının sadece gerekli olduğu kadar yetkiye sahip olması, veri bütünlüğünü korumaktadır ve izinsiz erişimi engellemektedir.

- (2) *Verimlilik:* Kullanıcıların görevlerine uygun rollere atanması, iş akışının daha verimli olmasını sağlamaktadır. Her kullanıcı kendi yetki alanında sorumluluk olarak sistemin etkin bir şekilde işlemlerini sağlamaktadır.
- (3) *Esneklik:* Roller ve kullanıcı yetkilerinin dinamik olarak yönetilebilmesi, sistemin ihtiyaçlara göre kolayca uyarlanabilmesini sağlamaktadır. Yeni kullanıcıların eklenmesi ve mevcut kullanıcıların rollerinin değiştirilmesi basit ve hızlı bir şekilde yapılabilmektedir.
- (4) *Şeffaflık ve İzlenebilirlik:* Kullanıcıların yaptıkları işlemler, atanmış rolleri sayesinde kolayca izlenebilmekte ve denetlenebilmektedir. Bu, hesap verebilirliği artırmakta ve olası hataların veya kötüye kullanımların önlenmesine yardımcı olmaktadır.

Böylece kullanıcıların ve rollerin bu şekilde tanımlanması sayesinde, sistemin genel performansı ve güvenliği arttırılmakta, kullanıcı deneyimini iyileştirilmekte ve iş akışları daha düzenli ve etkili bir şekilde yürütülebilmektedir.

4.1.7 Takımların Tanımlanması ve Yönetimi

Takım tanımlama ve takım yönetimi ekranı, arkeolojik veri yönetim sisteminde takımların oluşturulması, yönetilmesi ve kullanıcıların bu takımlara atanması için tasarlanmış bir arayüz sunmaktadır. Takım çalışmasının, arkeolojik saha çalışmalarında verimliliği ve koordinasyonu artırmak için kritik öneme sahip olması sebebiyle bu arayüz, kullanıcıların farklı takımlar oluşturmalarına, takım üyelerini belirlemelerine ve takım liderlerini atamalarına olanak tanımaktadır.



Şekil 4.6. Takım tanımlama ve takım yönetimi ekranı

Şekil 4.6.'da ekranın sol bölümünde mevcut takımlar listelenmiştir. Her takım, belirli bir amaç veya proje kapsamında faaliyet gösterebilmektedir. Yeni takımlar oluşturmak için 'Yeni Takım' butonu kullanılmaktadır. Mevcut takımlar üzerinde düzenleme yapmak veya silmek için ise ilgili butonlar kullanılmaktadır.

Ekranın orta bölümünde, seçili takımın üyeleri listelenmektedir. Takım üyeleri, rollerine göre farklı etiketlerle gösterilmektedir. Takım lideri, diğer üyelerden farklı olarak 'Lider' etiketi ile belirtilmektedir. Ekranın sağ bölümünde ise, mevcut kullanıcılar arasından takıma eklenebilecek kullanıcılar listelenmektedir.

4.1.7.1 Takım Lideri ve Yetkileri

Takım lideri, takımı yöneten ve takımın saha çalışmalarını koordine eden kişidir. Takım liderinin özel yetkileri şunlardır:

- *Veri Girişi Yetkisi:* Bir takım, bir buluntu keşfettiğinde, bu buluntunun sisteme girişini sadece takım lideri yapabilmektedir. Böylece, verilerin doğruluğu ve bütünlüğü garanti altına alınmaktadır. Ayrıca takım lideri, buluntunun tüm detaylarını sisteme girmekte ve ilgili kayıtları oluşturmaktadır.
- *Koordinasyon ve Yönetim:* Takım lideri, takım içi iletişimi ve iş birliğini artırmak amacıyla takım üyelerinin görevlerini uygulama üzerinden

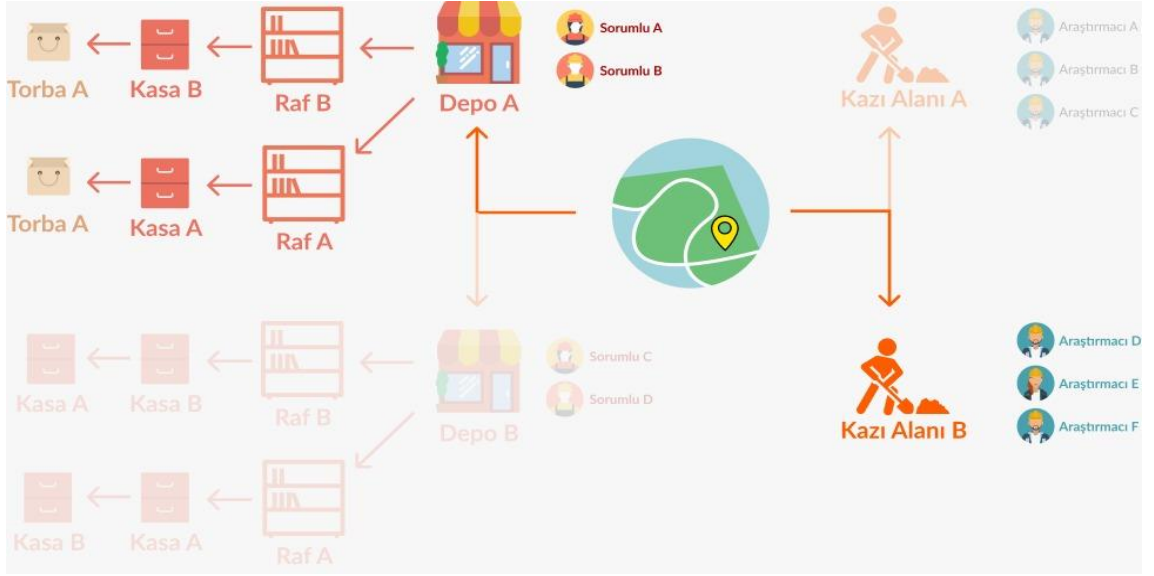
koordine etmekte ve çalışmaların düzenli bir şekilde ilerlemesini sağlamaktadır.

Takımların ve takım liderlerinin belirlenmesi, uygulamaya başlıca avantajlar sunmaktadır: Saha çalışmalarının daha organize ve koordine bir şekilde yürütülmesini sağlamaktadır. Her takım belirli bir görevi yerine getirmek için organize olmakta ve takım lideri tarafından yönlendirilmektedir. Böylece, saha çalışmalarının daha hızlı ve etkili bir şekilde tamamlanması sağlanmaktadır. Bununla birlikte, takımların dinamik olarak oluşturulması ve yönetilmesi, sistemin esnekliğini artırmakta; ihtiyaçlara göre yeni takımlar oluşturulabilmekte veya mevcut takımlar yeniden düzenlenebilmektedir. Bu işlem, sistemin değişen şartlara hızlıca uyum sağlamasını mümkün kılmaktadır. Takım çalışması ve liderlerin belirlenmesi, saha çalışmalarının izlenebilirliğini de artırmaktadır. Hangi takımın hangi buluntuları keşfettiği ve bu buluntuların sisteme nasıl girildiği kolayca takip edilebildiği için şeffaflık ve hesap verilebilirlik kolaylaşmaktadır.

4.1.8 Birim Şemalarının Tanımlanması ve Yönetimi

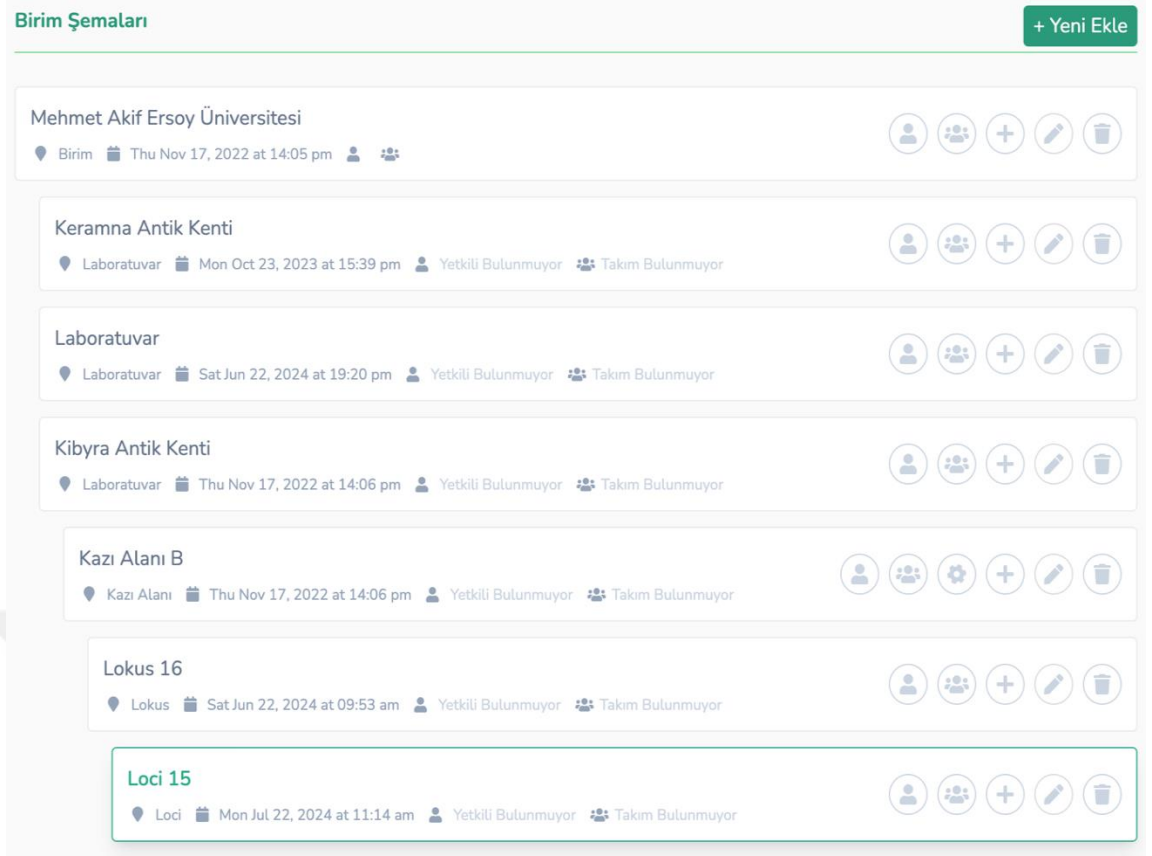
Birim şemalarının tanımlanması ve yönetiminin işleyişini örnek bir kazı alanı tanımlama modülü üzerinden görselleştirmek mümkündür. Şekil 4.7.'de yer alan kazı alanı modülü aşağıdaki özellikleri içermektedir:

- Birden fazla antik kazı alanı (Ör. Kibyra, Sagalassos) tanımlama,
- Her kazı alanına ait depo/depoları tanımlama ve depolarda bulunan her raf, kasa ve kutu/torba için barkodların üretimi,
- Kazı alanında çalışmaların yürütüldüğü bölgelerin tanımlanması,
- Kazı alanı bölgelerinde çalışan lider ve araştırmacıların tanımlanması.



Şekil 4.7. Kazı alanı işleyiş modülleri

Arkeolojik veri yönetim sisteminde, birimlerin ve alt birimlerin hiyerarşik olarak tanımlanması ve yönetilmesi, veri organizasyonu ve erişilebilirliği açısından büyük önem taşımaktadır. Birim şemalarının tanımlanması ve yönetilmesi ekranı (Şekil 4.8) birimlerin ve bu birimlerin altına gelebilecek yapılarının tanımlanmasını ve yönetilmesini sağlamaktadır.



Şekil 4.8. Birim şemalarının tanımlanması ve yönetilmesi ekranı

4.1.8.1 Ana Birimlerin Tanımlanması

Şekil 4.8.’de yer alan ekranın sol bölümünde mevcut birim şemaları listelenmiştir. Her birim, kullanıcılar ve takımlar tarafından yönetilebilmektedir. Yeni birim şemaları oluşturmak için ‘Yeni Ekle’ butonu kullanılmaktadır. Mevcut birimler üzerinde düzenleme yapmak veya silmek için ilgili butonlar kullanılmaktadır.

Ekranın orta bölümünde, seçili birime ait alt yapılar listelenmektedir. Bu yapılar, belirli bir düzen ve hiyerarşi içinde organize edilmektedir. Sağ bölümde ise, mevcut yapılar arasından seçilerek birime eklenebilecek yeni yapılar listelenmektedir.

4.1.8.2 Alt Yapıların Tanımlanması

Birim şemaları ekranında, yeni bir yapı eklemek için sağ bölümdeki “+” butonu kullanılmaktadır. Açılan pencerede, seçili birimin altına eklenebilecek yapı türleri listelenmektedir. Bu yapı türleri, önceki tanımlamalara göre belirlenmiş olup, kullanıcıların birim şemasını doğru ve tutarlı bir şekilde oluşturmasını sağlamaktadır.

Yeni Yapı

Mehmet Akif Ersoy Üniversitesi (Birim) altına eklenecek yeni yapıyı seçiniz.

Tür Seçiniz

Depo Kazı Alanı Laboratuvar

Müze Raf

Barcode

2964058173

Lokasyon (GPS Koordinatını virgül ile ayırarak giriniz)

...

Ad

...

Açıklama

...

Birim için gerekli bir açıklama/hot

İPTAL KAYDET

Şekil 4.9. Birim Şemasına Yeni Yapı Ekleme Ekranı

Örneğin, ‘Mehmet Akif Ersoy Üniversitesi’ biriminin altına ‘Laboratuvar’, ‘Depo’, ‘Kazı Alanı’ veya ‘Raf’ gibi alt yapılar eklenebilmektedir. Birim şemasının hiyerarşik ve düzenli bir şekilde oluşturulması için kullanıcı, eklemek istediği yapıyı seçtikten sonra, yapı ile ilgili detaylı bilgileri (ör. barkod, lokasyon, ad, açıklama) girmekte ve yapıyı kaydetmektedir. Oluşturulan yapının sunduğu başlıca avantajlar bulunmaktadır:

- *Hiyerarşik Düzen ve Organizasyon:* Yapıların hiyerarşik olarak tanımlanması ve organize edilmesi, verilerin düzenli ve sistematik bir şekilde yönetilmesini sağlamaktadır.
- *Esneklik ve Uyarlanabilirlik:* Yeni yapıların eklenmesi ve mevcut yapıların düzenlenmesi, sistemin esnekliğini ve uyarlanabilirliğini artırmaktadır. Kullanıcılar, ihtiyaçlarına göre birim şemasını kolayca güncelleyebilmektedir.
- *Veri Doğruluğu ve Tutarlılığı:* Alt yapıların belirli kurallar ve hiyerarşiler doğrultusunda eklenmesi, veri girişinde doğruluğu ve tutarlılığı sağlamaktadır.
- *Kolay ve Hızlı Erişim:* Hiyerarşik düzen sayesinde, kullanıcılar veriye daha hızlı ve kolay erişebilmekte; iş süreçlerini hızlandırılmasıyla kullanıcı deneyimi iyileştirilmektedir.

- *Görselleştirme ve Analiz:* Birim şemalarının görsel olarak organize edilmesi, verilerin daha iyi anlaşılmasını ve analiz edilmesine imkân tanımaktadır.

4.1.9 Birimlere Yetkili veya Takımların Tanımlanması

Arkeolojik veri yönetim sisteminde, birimlerin etkin ve güvenli bir şekilde yönetilmesi için yetkili kişilerin ve takımların tanımlanması kritik bir rol oynamaktadır. Bu başlık altında, birimlere atanacak yetkililerin ve takımların nasıl tanımlandığı ve bu sürecin sağladığı avantajlar ele alınacaktır.

4.1.10 Yetkili Kişilerin Tanımlanması;

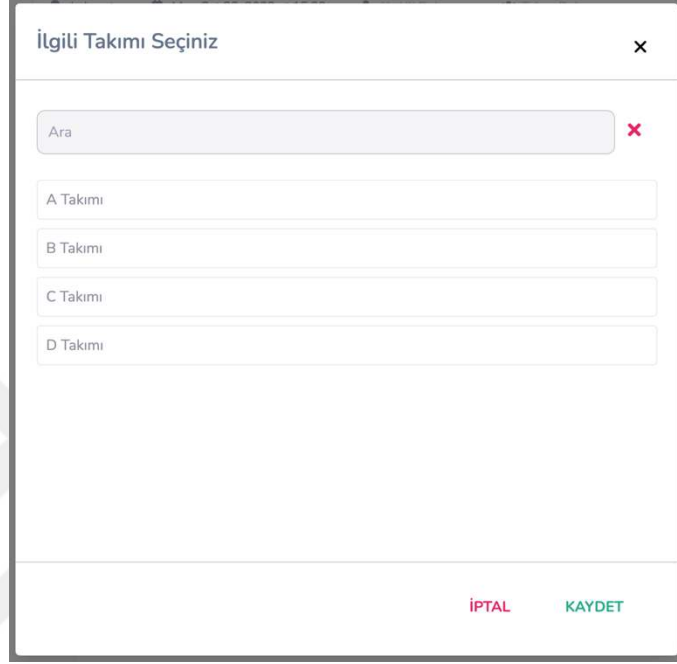
Birimler, belirli bir yetki çerçevesinde yönetilmelidir. Bu amaçla, yetkili kişilerin belirlenmesi ve atanması süreci aşağıdaki gibi gerçekleştirilmektedir.

Şekil 4.10. Yöneticiler için birim kullanıcısı tanımlama ekranı

Bu ekranda, yetkili kişilerin atanması için kullanıcıların listelendiği bir arayüz bulunmaktadır. Bu arayüzde, mevcut kullanıcılar arasından uygun yetkiye sahip kişiler seçilmekte ve ait olduğu birime atanmaktadır. Seçilen yetkililerin sistemde sahip oldukları roller (Süper Admin, Admin, Normal Kullanıcı, İzleyici, vb.) göz önünde bulundurularak, uygun yetki seviyeleri belirlenmektedir. Bu sayede, her birim için doğru ve gerekli yetkilere sahip kişiler atanmış olur.

4.1.11 Takımların Tanımlanması

Birimlerde koordineli çalışmalar yürüten ve belirli projeler veya görevler için oluşturulmuş takımlar, birimlerin yönetiminde önemli rol oynamaktadır. Bu sebeple, oluşturulan takımların birimlere atanması için ayrı bir ekran tasarlanması ihtiyacı doğmuştur.



Şekil 4.11. Takımların birimlere atanma ekranı

Takımların birimlere atanma ekranında yapılabilecek işlemler ve işlevleri aşağıda sunulmuştur:

- *Takım Seçimi:* Birime atanacak takımlar, mevcut takım listesinden seçilmektedir. Her takım, belirli bir görev veya proje kapsamında organize olmuştur ve ilgili birimle uyumlu olacak şekilde atanmaktadır.
- *Lider ve Üyeler:* Takım lideri ve üyeleri, takımın iç yapısını oluşturmaktadır. Takım lideri, takımın koordinasyonundan sorumludur ve en yetkili kişi olarak atanır.
- *Takım Arama ve Filtreleme:* Takım atama arayüzünde, mevcut takımlar arasında hızlı arama ve filtreleme yapılabilir. Bu özellik, doğru takımın hızlı bir şekilde bulunmasını ve atanmasını sağlar.

4.1.12 Kullanıcı Arayüzü: Yetkilendirilmiş Kullanıcılar için İşlem Menüsü

Yazılımda, tüm sistem tanımlamaları ve yetkilendirmeler yapıldıktan sonra, kullanıcıların erişip işlemler yapabileceği bir arayüz sağlanmıştır. Bu arayüz, kullanıcıların sisteme erişimini, verileri yönetmesini ve çeşitli işlemleri gerçekleştirmesini sağlayan menülerden oluşmaktadır. Bu bölümde, yetkilendirilmiş kullanıcıların erişebileceği işlem menüsü ve bu menünün sunduğu işlevler sırası ile açıklanmıştır.



Şekil 4.12. Kullanıcı arayüz ekranı

4.1.13 Buluntu Ekleme Menüsü

Bu menü seçeneği, kullanıcıların sisteme yeni buluntular eklemelerine olanak tanımaktadır. Kullanıcılar, bu seçenek aracılığıyla yeni veri girişi yapmakta ve bu verileri sistemde doğru bir şekilde kayıt altına almaktadır. Buluntu ekleme menüsünde buluntunun 'bulunduğu yer, buluntu durumu, buluntu türü, buluntu rengi/renkleri, buluntu ekleme detayları, buluntu imajı ve coğrafi bilgi'lere dair bilgilerin girilmesini sağlayan alt menüler bulunmaktadır.

Bulunduğu Yer: Yeni eklenen buluntunun hangi obje veya yer altına eklenmesi gerektiğini belirlemek için kullanılmaktadır. Kullanıcılar, mevcut yerler arasından seçim yaparak buluntunun konumunu belirlemektedir. Veri girişinin doğru bir şekilde organize edilmesi için seçenekler arasında kazı alanları, laboratuvarlar, lokuslar ve depolar yer almaktadır.

Ait Olduğu Yer

Arama...

Seçim

x Loci 15

- Kibyra Antik Kenti - 54247811098140992284

- Kazı Alanı B - 34160200454254272261

- Lokus 16 - 04843568611035866308

Loci 15 -

+ Depo A - 54886996137553953254

Şekil 4.13. Buluntu yeri kayıt

Buluntu Durumu: Buluntunun mevcut durumunu belirlemek için kullanılmaktadır. Kullanıcılar, buluntunun durumunu seçerek sisteme kaydetmektedir. Buluntu durumu kaydedilirken buluntunun fiziksel durumu hakkında bilgiler seçilmektedir.

Buluntu Durumu

Arama...

Seçili Buluntu Durumu

x Hafif Hasarlı

Çok İyi Durumda

Hafif Hasarlı

Çok Hasarlı

Eksik Parçalı

Şekil 4.14. Buluntu durumu kaydı

Buluntu Türü: Eklenen buluntunun türünü belirlemek için kullanılmaktadır. Kullanıcılar, mevcut türler arasından seçim yaparak buluntunun kategorisini belirlemektedir. Türlerin kayıt altına alınması, buluntuların sınıflandırılmasını ve analiz edilmesini kolaylaştırmaktadır.

Buluntu Türü

Arama...

Seçili Buluntu Türleri

x Cam Kaplar ve Şişeler x Boncuklar ve Süs Eşyaları

+ Cam ve Porselen

+ Kağıt ve Parşömen

+ Metaller

+ Organik Malzemeler

+ Seramik

+ Taş Eserler

Şekil 4.15. Buluntu türü kaydı

Buluntu Rengi/Renkleri: Kullanıcıların buluntunun renk bilgilerini girdiği alandır ve kullanıcılar, buluntunun sahip olduğu renkleri seçerek sisteme kaydetmektedir. Bu bilgilerin girilmesi, buluntuların ileride daha detaylı analiz edilmesine ve sınıflandırılmasına yardımcı olacaktır.

Şekil 4.16. Buluntu rengi kaydı

Buluntu Detayı Ekleme: Bu bölüm, buluntuların görselleri, fiziksel özellikleri, coğrafi koordinatları ve açıklamaları gibi çeşitli bilgilerin girileceği ekranı temsil etmektedir.

Şekil 4.17. Buluntuya dair detay bilgilerin kaydı

Şekil 4.17’de yer alan buluntu *imajı* sekmesi, buluntunun görselini sisteme yüklemek için kullanılmaktadır. Eklenen görsel, aramalarda ve listelerde görüntülenmektedir. Buluntuya dair fiziksel özelliklerin girilmesi için ağırlık, boy, en ve yükseklik sekmelerine veri girişi yapılmaktadır.

Coğrafi Bilgi: Buluntunun bulunduğu yerin coğrafi koordinatları bu alana virgülle ayrılmış olarak girilmektedir.

Açıklama (Not): Buluntu hakkında detaylı açıklamalar ve notlar bu alana girilmektedir. Bu açıklamalar, buluntunun tarihi, durumu ve diğer önemli bilgileri içerebilmektedir. Örneğin; ‘Neandertal ailesine ait kemikler; 3 yetişkin kadın, 3 yetişkin erkek, 3 genç ve 1 bebeğe ait olduğu saptanan kemiklerin hepsi çiğnenme, kemirilme ve sıyrılma izleri taşımaktadır.’ şeklinde buluntuyu betimleyecek ayrıntılı bilgilere yer verilmektedir.

Buluntu detayı ekleme menüsünün kullanıcıya sağladığı avantajlar şu şekilde sıralanabilir:

- *Detaylı Veri Girişi:* Kullanıcıların, buluntuların fiziksel özelliklerini, görsellerini ve coğrafi bilgilerini detaylı bir şekilde sisteme girebilmeleri sayesinde, veriler daha organize ve analiz edilebilir hale gelmektedir.
- *Görsel Belgelenme:* Buluntuların görsellerinin sisteme yüklenmesi, görsel referans sağlamak ve verilerin daha iyi anlaşılmasına yardımcı olmaktadır.
- *Kolay Erişim ve Analiz:* Detaylı açıklamalar ve notlar, kullanıcıların buluntulara dair bilgileri kolayca erişip analiz etmelerine olanak tanımaktadır.

4.1.14 Buluntu Arama

Buluntuların etkili bir şekilde aranması ve bulunması büyük önem taşımaktadır. Bu sistem, kullanıcıların buluntuları çeşitli kriterlere göre aramalarını sağlayan iki temel arama yöntemini desteklemektedir: *Özelliklere Göre Arama* ve *QR Kod ile Arama*.

4.1.15 QR Kod ile Arama

QR Kod ile arama, kullanıcıların buluntulara hızlı ve doğru bir şekilde erişmelerini sağlamaktadır. Bu yöntem, veri girişinde yapılan hataları azaltmaktadır.

Buluntu Arama

Barkod ile Ara Özelliklere Göre Arama


NO : 57082282744026382242



ID	: 1
Barcode	: 57082282744026382242
İlk Eklenme Tarihi	: 18.07.2024 11:55:21
Son İşlem Tarihi	: 18.07.2024 11:55:21
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Yeni Eklendi
Yapılan İşlem	deneme
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Lokus 16
Durumu	: Çok İyi Durumda
Tür	: Kaplar, Tabaklar ve Kaseler, Seramik Parçaları,
Renk	: Kırmızı, Mavi, Sarı,
Boyut	: X:40 Y:40 X:40

Tüm Yapılanları Göster

Şekil 4.18. QR ile buluntu arama

4.1.16 Özelliklere Göre Arama

Kullanıcıların, buluntuları çeşitli kriterlere göre detaylı bir şekilde aramalarını sağlamaktadır. Bu yöntem, kullanıcılara buluntuların belirli özelliklerine göre filtreleme yaparak istedikleri verilere hızlıca ulaşma imkânı sunmaktadır.

Buluntu Arama

Barkod ile Ara Özelliklere Göre Arama

Şuan Bulunduğu Yer

Kıbrıs Antik Kenti - 54247811098140992284 x

Buluntu Türü

Hafif Hasarlı x

Buluntu Renkleri

Kırmızı x Kahverengi x Beyaz x

Buluntu Durumu

Hafif Hasarlı x

Buluntuda Yapılan İşlem Türü

Adım Silindi x Bilgileri Düzenlendi x Buluntuyu Silindi x Görsel Eklendi x Restorasyon Yapıldı x Transfer Başlatıldı x Transfer Edildi x
Video Eklendi x Yeni Eklendi x

ARA

Şekil 4.19. Özelliklere göre buluntu arama ekranı

Buluntu arama ekranında, özelliklere göre buluntu arama aşağıda yer alan kriterleri içermektedir:

- *Mevcutta Bulunduğu Yer:* Belirli bir alanda bulunan tüm buluntuların listelenebilmesi için kullanıcılar, buluntunun mevcut bulunduğu yeri seçerek arama yapabilmektedir. Örneğin; Kıbrıs Antik Kenti.
- *Buluntu Türü:* Kullanıcılar, belirli bir türdeki buluntuları listeleyebilmek için buluntunun türüne göre arama yapabilmektedir. Örneğin; Hafif Hasarlı.

- *Buluntu Renkleri:* Sadece belirli renklerdeki buluntuları görmek için kullanıcılar, buluntunun sahip olduğu renkleri seçerek arama yapabilmektedir. Örneğin; Kırmızı, Kahverengi, Beyaz.
- *Buluntu Durumu:* Kullanıcılar, belirli bir durumda olan buluntuları listeleyebilmek için buluntunun fiziksel durumuna göre arama yapabilmektedir. Örneğin; Hafif Hasarlı.
- *Buluntuda Yapılan İşlem Türü:* Kullanıcılar, belirli türde işlemler yapılmış buluntuları görebilmek için buluntular üzerinde yapılan işlemlere göre arama yapabilmektedir. Örneğin; Restorasyon Yapıldı.

Sonuç olarak özelliklere göre yapılan arama, kullanıcıların belirli kriterlere göre detaylı filtrelemeler yaparak istedikleri buluntulara hızlıca ulaşmalarını sağlamaktadır.

4.1.17 Yapay Zekâ ile Buluntu Arama

‘Yapay Zekâ ile Buluntu Arama’ bölümü, ARKHESTOR arkeolojik veri yönetim sisteminde kullanıcıların benzer buluntuları hızlı ve doğru bir şekilde bulmalarını sağlayan ileri seviye bir araç sunmaktadır. Bu özellik, veri analizini ve bilimsel araştırmaları önemli ölçüde iyileştirmekte kullanıcıların zamanını verimli bir şekilde yönetmelerine olanak tanımaktadır.



Yeni Arama



Benzerlik Puanı	: 97.73
ID	: 21
Barcode	: 71351315287845905202
İlk Eklenme Tarihi	: 01.08.2024 11:55:21
Son İşlem Tarihi	: 31.07.2024 11:27:38
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Transfer Başlatıldı
Yapılan İşlem	deneme
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Loci 15
Durumu	:
Tür	: Heykeller,
Renk	: Kırmızı, Mavi, Sarı,
Boyut	: X:50 Y:40 X:55



Benzerlik Puanı	: 66.40
ID	: 4
Barcode	: 92880617587456022891
İlk Eklenme Tarihi	: 01.08.2024 11:55:21
Son İşlem Tarihi	: 18.07.2024 11:55:48
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Bilgileri Düzenlendi
Yapılan İşlem	deneme
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Lokus 16
Durumu	: Çok İyi Durumda
Tür	: Kaplar,
Renk	: Kırmızı, Mavi, Sarı,
Boyut	: X:40 Y:40 X:40



Benzerlik Puanı	: 66.40
ID	: 5
Barcode	: 68993684850632961403
İlk Eklenme Tarihi	: 01.08.2024 11:55:21
Son İşlem Tarihi	: 18.07.2024 11:55:54
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Bilgileri Düzenlendi
Yapılan İşlem	deneme
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Lokus 16
Durumu	: Çok İyi Durumda
Tür	: Kaplar,
Renk	: Kırmızı, Mavi, Sarı,
Boyut	: X:40 Y:40 X:40



Benzerlik Puanı	: 58.03
ID	: 16
Barcode	: 03638348522882904601
İlk Eklenme Tarihi	: 01.08.2024 11:55:21
Son İşlem Tarihi	: 27.07.2024 11:29:50
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Yeni Eklendi
Yapılan İşlem	deneme
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Loci 15
Durumu	: Hafif Hasarlı
Tür	: Kaplar,
Renk	: Mavi, Kırmızı, Kahverengi,
Boyut	: X:50 Y:20 X:50

Şekil 4.20. Yapay zekâ destekli buluntu arama ekranı

Örneğin arkeolog, belirli bir buluntuya benzer diğer buluntuları aramak istediğinde, ‘Yapay Zekâ ile Buluntu Arama’ bölümünü kullanarak yeni bir arama başlatmaktadır. Yapay zekâ, seçilen görseli analiz etmekte ve benzer özelliklere sahip diğer buluntuları listelemektedir. Her bir sonuç, benzerlik puanı ile birlikte detaylı bilgiler içermekte, bu da arkeoloğun ilgili buluntular hakkında kapsamlı bilgi edinmesini sağlamaktadır.

4.1.18 Buluntu Geçmişi

‘Buluntu Geçmişi’ ekranında kullanıcılar, buluntuların son 1 gün, son 2 gün, son 3 gün ve son 7 gün gibi zaman filtreleme seçeneklerini kullanarak belirli bir zaman aralığında eklenen veya üzerinde işlem yapılan buluntuları görüntüleyebilmektedirler.

Buluntu Geçmişi

Son 1 GünSon 2 GünSon 3 GünSon 7 Gün



ID	: 11
Barcode	: 31392180125211777938
İlk Ekleme Tarihi	: 18.07.2024 17:47:31
Son İşlem Tarihi	: 18.07.2024 17:47:31
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Transfer Başlatıldı
Yapılan İşlem	
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Lokus 16
Durumu	: Çok İyi Durumda
Tür	: Kaplar, Tabaklar ve Kaseler, Seramik Parçaları,
Renk	: Kırmızı, Mavi, Sarı,
Boyut	: X:40 Y:40 X:40

Tüm Yapılanları Göster



ID	: 10
Barcode	: 16494888239377071169
İlk Ekleme Tarihi	: 18.07.2024 17:33:13
Son İşlem Tarihi	: 18.07.2024 17:33:13
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Yeni Eklendi
Yapılan İşlem	deneme
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Lokus 16
Durumu	: Çok İyi Durumda
Tür	: Kaplar, Tabaklar ve Kaseler, Seramik Parçaları,
Renk	: Kırmızı, Mavi, Sarı,
Boyut	: X:40 Y:40 X:40

Tüm Yapılanları Göster



ID	: 9
Barcode	: 44245390451529233275
İlk Ekleme Tarihi	: 18.07.2024 13:09:56
Son İşlem Tarihi	: 18.07.2024 13:09:56
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Yeni Eklendi
Yapılan İşlem	deneme
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Lokus 16
Durumu	: Çok İyi Durumda
Tür	: Kaplar, Tabaklar ve Kaseler, Seramik Parçaları,
Renk	: Kırmızı, Mavi, Sarı,
Boyut	: X:40 Y:40 X:40

Tüm Yapılanları Göster

Şekil 4.21. Buluntu geçmişi ekranı

Buluntu geçmişi ekranında, her bir buluntu için yapılan işlemler detaylı olarak listelenmektedir. Bu işlemler, buluntunun zaman içinde nasıl değiştiğini ve üzerinde ne tür işlemler yapıldığını göstermektedir. Bu ekran yer alan örnek işlemler şunlardır: Yeni Eklendi, Bilgileri Düzenlendi, Transfer Başlatıldı, Restorasyon Yapıldı, Görsel Eklendi, Video Eklendi, Adım Silindi, Buluntuyu Silindi, Transfer Tamamlandı.

Buluntu Geçmişi

Son 1 GünSon 2 GünSon 3 GünSon 7 Gün



ID	: 11
Barcode	: 31392180125211777938
İlk Ekleme Tarihi	: 18.07.2024 17:47:31
Son İşlem Tarihi	: 18.07.2024 17:47:31
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Transfer Başlatıldı
Yapılan İşlem	
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Lokus 16
Durumu	: Çok İyi Durumda
Tür	: Kaplar, Tabaklar ve Kaseler, Seramik Parçaları,
Renk	: Kırmızı, Mavi, Sarı,
Boyut	: X:40 Y:40 X:40

Tüm Yapılanları Göster

Buluntu Üzerinde Yapılan İşlemler

+ Yeni

İşlem Türü	Yeni Eklendi	Açıklama/Not
Yapan Kişi	Can YASTIOĞLU	deneme
Oluşturma Tarihi	18.07.2024 17:47:31	
İşlem Türü	Restorasyon Yapıldı	Açıklama/Not
Yapan Kişi	Can YASTIOĞLU	Temizleme işlemi yapıldı. Kırık parçalar birleştirildi.
Başlama Tarihi	21.07.2024 21:50:00	
Bitirme Tarihi	23.07.2024 21:50:00	
Oluşturma Tarihi	22.07.2024 22:58:42	
İşlem Türü	Transfer Başlatıldı	Açıklama/Not
Yapan Kişi	Can YASTIOĞLU	
Oluşturma Tarihi	22.07.2024 22:59:11	

Şekil 4.22. Buluntu üzerinde yapılan işlemler ekranı

Yine aynı ekranda mevcut buluntular üzerine yeni işlemler eklenebilmektedir. Bu işlemler arasında görsel ekleme, video ekleme, restorasyon ekleme ve transfer etme gibi çeşitli işlemler yer almaktadır.

Yeni İşlem

Lütfen Yapmak istediğiniz işlem türünü seçiniz

Görsel Ekle

Restorasyon Ekle

Transfer Et

Video Ekle

İPTAL

KAYDET

Şekil 4.23. Buluntu yeni işlem ekranı

4.1.19 Transfer İşlemleri

Buluntuların bir birimden başka bir birime transfer edilmesi sürecini yönetmek için ‘Transfer İşlemleri’ ekranı kullanılmaktadır. Bu ekran, kullanıcıların buluntu transfer süreçlerini kolay ve etkili bir şekilde yönetmelerini sağlamakta; transfer işlemleri gönderilen, alınan, tamamlanan ve reddedilen transferleri kapsamaktadır.

Transfer İşlemleri

Gönderdikleriniz **Aldıklarınız** **Tamamlananlar** **Reddedilenler**

Birimize Gönderilen Buluntular



ID	: 10
Barcode	: 16494888239377071169
İlk Ekleme Tarihi	: 24.07.2024 10:31:52
Son İşlem Tarihi	: 18.07.2024 17:33:13
Ekleyen	: Can YASTIOĞLU

İşlem Türü	Transfer Başlatıldı
Yapılan İşlem	Önemli bir buluntu. Ekip gelene kadar depoda bekletilsin.
Yapan Kişi	: Can YASTIOĞLU
Bulunduğu Yer	: Lokus 16
Durumu	: Çok İyi Durumda
Tür	:
Renk	:
Boyut	: X:40 Y:40 X:40

Tüm Yapılanları Göster

Şekil 4.24. Transfer işlemleri ekranı

Ekranın üst kısmında, transfer işlemlerinin durumuna göre farklı sekmeler bulunmaktadır. Bu sekmeler, kullanıcıların transfer süreçlerini kolayca takip etmelerini sağlamaktadır:

- Gönderdikleriniz:** Bu sekmede, kullanıcının başka birimlere gönderdiği tüm buluntular listelenmektedir. Kullanıcılar, gönderilen buluntuların durumunu ve transfer sürecini bu ekrandan takip edebilmektedir.
- Aldıklarınız:** Bu sekmede, kullanıcının başka birimlerden aldığı buluntular listelenmekte ve kullanıcılar, alınan buluntuların detaylarını ve transfer sürecini görebilmektedir.
- Tamamlananlar:** Bu sekmede, tamamlanan tüm transfer işlemleri listelenmektedir. Kullanıcılar, başarılı bir şekilde tamamlanmış transferlerin detaylarını bu sekmeden inceleyebilmektedir.
- Reddedilenler:** Reddedilen transfer işlemleri listelendiği bu sekmede kullanıcılar, reddedilen transferlerin nedenlerini ve detaylarını görebilmektedir.

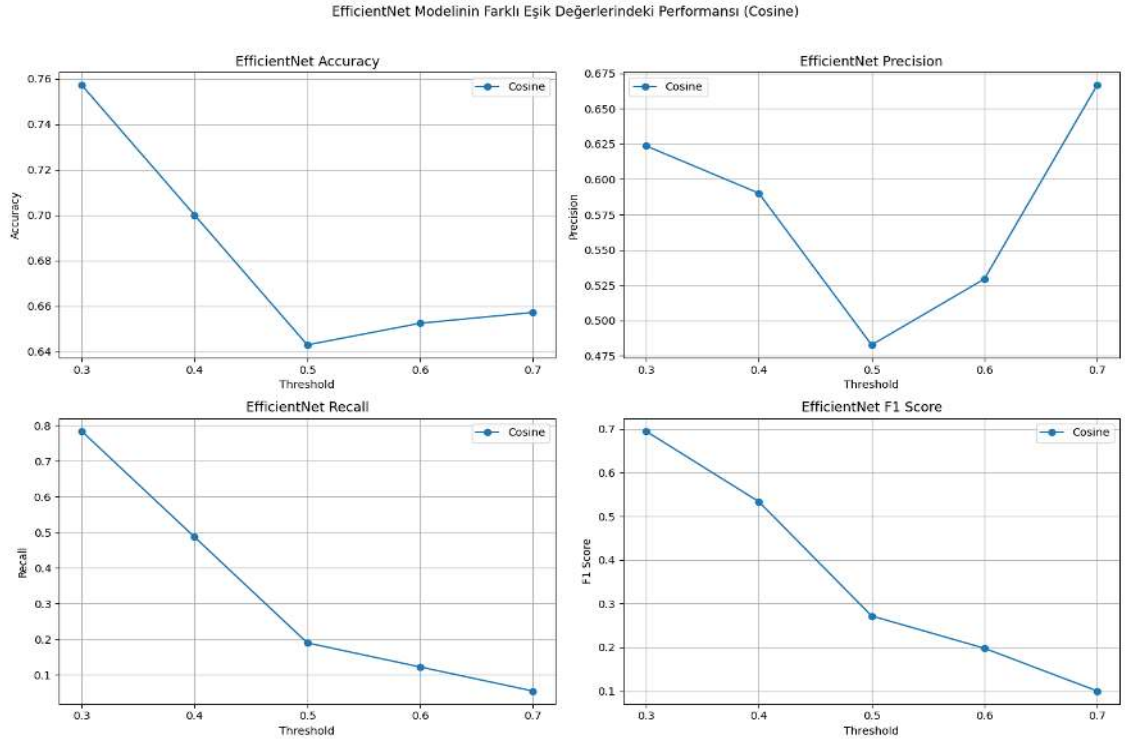
Transfer işlemleri ekranı, yine kullanıcılara bazı başlıca avantajlar sunmaktadır. Transfer işlemleri ekranı, kullanıcıların buluntu transferlerini kolayca takip etmelerini ve yönetmelerini sağlamakta ve transfer süreçlerinin etkinliğini artırmaktadır. Her bir transfer işlemi için detaylı bilgilerin sunulması, kullanıcıların transfer süreçlerini daha iyi anlamalarını sağlamaktadır. Transfer işlemlerinin durumlarına göre farklı sekmelerde organize edilmesi, kullanıcıların transfer süreçlerini düzenli bir şekilde takip etmelerine olanak tanımaktadır.

Sonuç olarak, ‘Transfer İşlemleri’ ekranı, arkeolojik veri yönetim sisteminde buluntu transfer süreçlerinin etkin bir şekilde yönetilmesini sağlamaktadır. Kullanıcılar, bu ekran aracılığıyla transfer edilen buluntuları detaylı bir şekilde görebilmekte, takip edebilmekte ve yönetebilmektedir.

4.1.19.1 Ön Eğitimli Modeller ile Benzerlik Fonksiyonlarının Performans Kıyaslaması

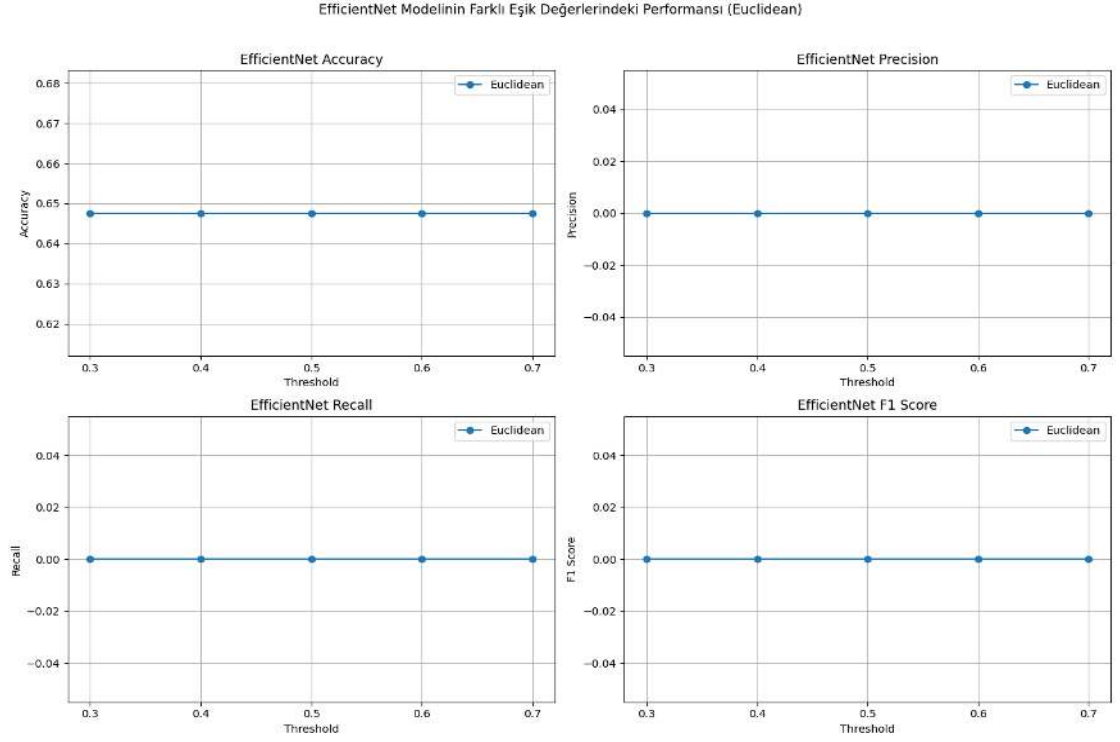
Bu başlık altında performans kıyaslaması, EfficientNetB0, ResNet50, VGG16 ve InceptionV3 gibi popüler ön eğitimli modellerin ve Cosine Similarity, Jaccard, Overlap, Euclidean ve Pearson benzerlik fonksiyonunun arkeolojik buluntuların sınıflandırılması ve benzerlik analizinde ne kadar etkili olduğuna dair değerlendirmeler yapılacaktır.

Kıyaslamalar, doğruluk oranları, hesaplama süreleri ve kaynak verimliliği gibi ölçütler üzerinden gerçekleştirilecektir. Bu analizler sonucunda, ARKHESTOR sistemi için en uygun model ve benzerlik fonksiyonu kombinasyonu belirlenecek ve bu kombinasyonun sunduğu avantajlar detaylı bir şekilde incelenecektir.



Şekil 4.25. EfficientNet modelinin farklı eşik değerlerindeki ile performans sonuçları (Cosine)

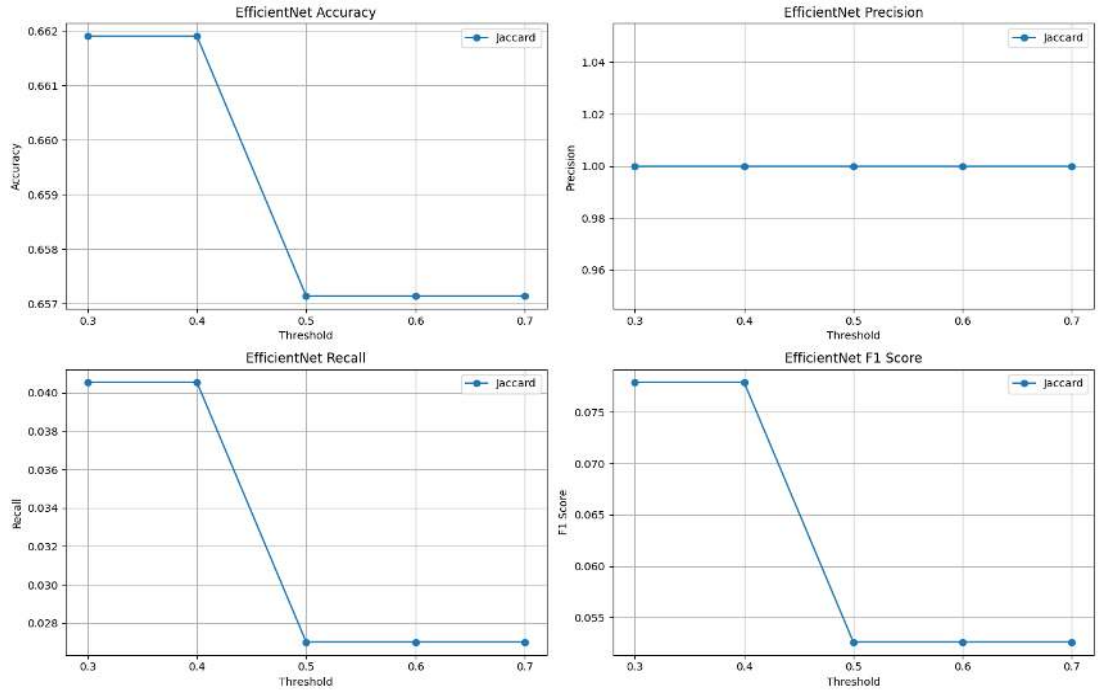
EfficientNet modelinin farklı eşik değerlerindeki performansını göstermekte ve modelin eşik değerine duyarlılığını ortaya koymaktadır. 0,3 eşik değerinde doğruluk, kesinlik, duyarlılık ve F1 skoru gibi metriklerde en yüksek performans gözlemlenmekte, ancak eşik değeri arttıkça bu performanslar düşüşe geçmektedir. Özellikle, eşik değeri 0,5'te belirgin bir düşüş yaşanırken, 0,7'de bazı metriklerde kısmi toparlanmalar görülmektedir. Bu durum, modelin performansının belirli eşik değerlerinde optimize edilmesinin önemini vurgulamakta ve 0,3 eşik değerinin genel olarak en iyi sonuçları verdiğini göstermektedir.



Şekil 4.26. EfficientNet modelinin farklı eşik değerlerdeki ile performans sonuçları (Euclidean)

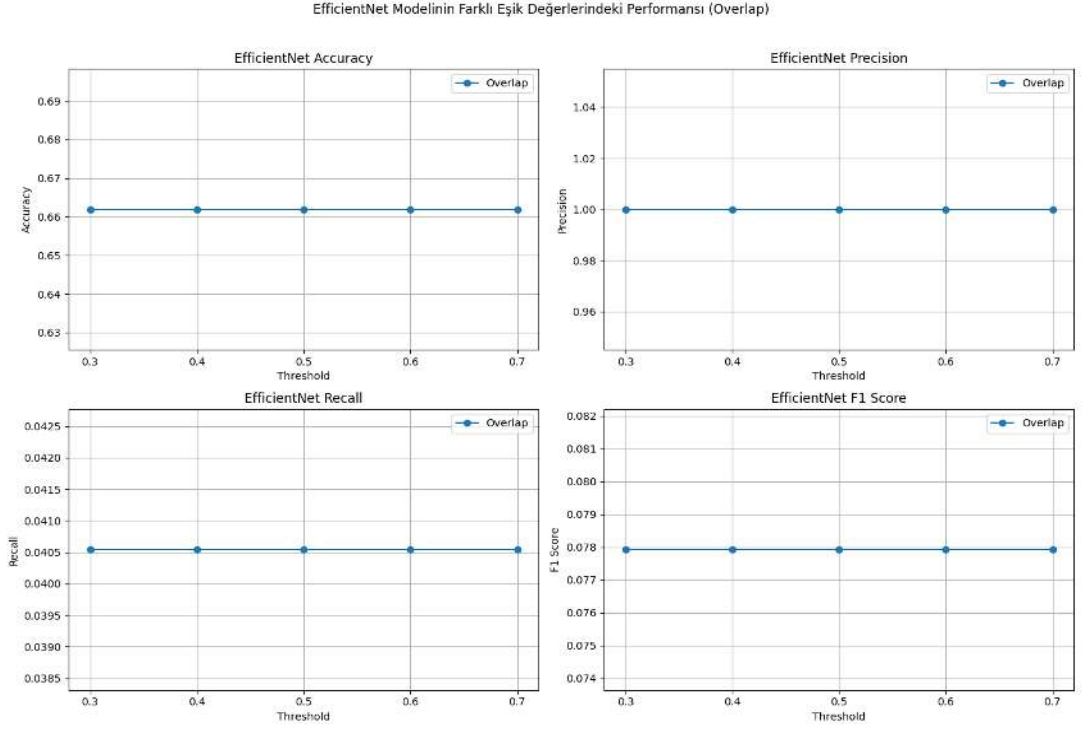
EfficientNet modelinin Euclidean fonksiyonu değerlendirildiğinde eşik değeri değişimlerine karşı duyarsız kaldığını ve performansın sabit kaldığını göstermektedir. Modelin bu metriğe göre stabilize bir performans sunmasından ve belirli bir eşik değeri seçiminin performansı etkilemediğinden dolayı ‘ARKHESTOR’ yazılımı için uygun olmadığına karar verilmiştir.

EfficientNet Modelinin Farklı Eşik Değerlerindeki Performansı (Jaccard)



Şekil 4.27. EfficientNet modelinin farklı eşik değerlerdeki ile performans sonuçları (Jaccard)

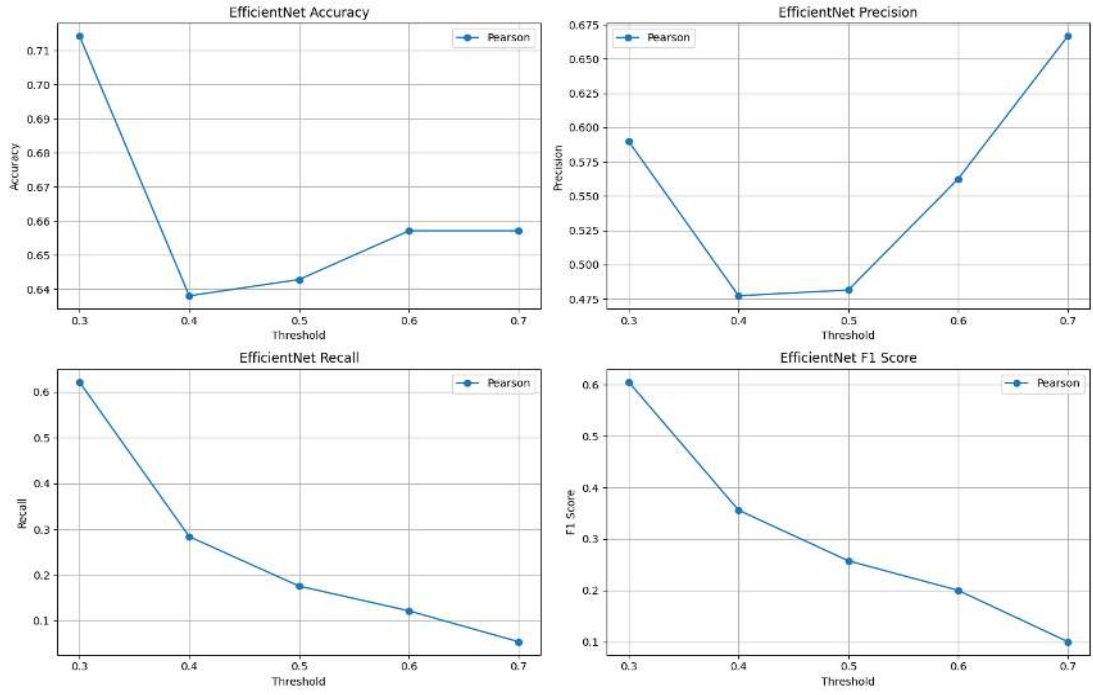
EfficientNET modelinin Jaccard fonksiyonunu kullanmak uygun olabilir, ancak yalnızca belirli bir eşik değeri aralığında, yani 0,4 veya altındaki değerlerde performans düşüşü, modelin güvenilirliğini ve genel doğruluğunu olumsuz etkilediği görülmektedir. Bu nedenle ‘ARKHESTOR’ yazılımı için uygun olmadığına karar verilmiştir.



Şekil 4.28. EfficientNet modelinin farklı eşik değerlerdeki ile performans sonuçları (Overlap)

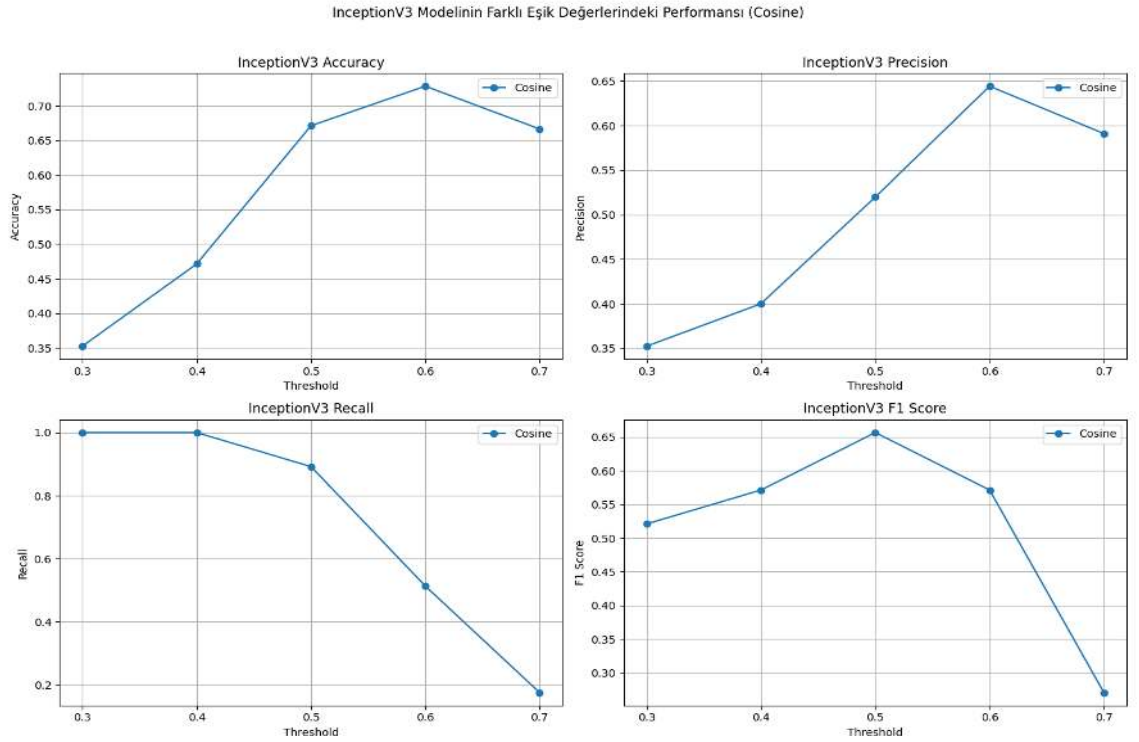
EfficientNet modeli ile Overlap benzerlik fonksiyonunu kullanmak, performansın tüm eşik değerlerinde sabit kaldığı düşünüldüğünde uygun olabilir. Ancak, bu sabit performans, modelin esnekliğini ve farklı eşik değerlerine adaptasyon kabiliyetini sınırlayabileceği görülmektedir. Bu nedenle, modelin performansını optimize etmek ve belirli durumlarda iyileştirmek hedeflendiğinde, Overlap benzerlik fonksiyonu kısıtlayıcı olabileceğinden dolayı ‘ARKHESTOR’ yazılımı için uygun olmadığına karar verilmiştir.

EfficientNet Modelinin Farklı Eşik Değerlerindeki Performansı (Pearson)



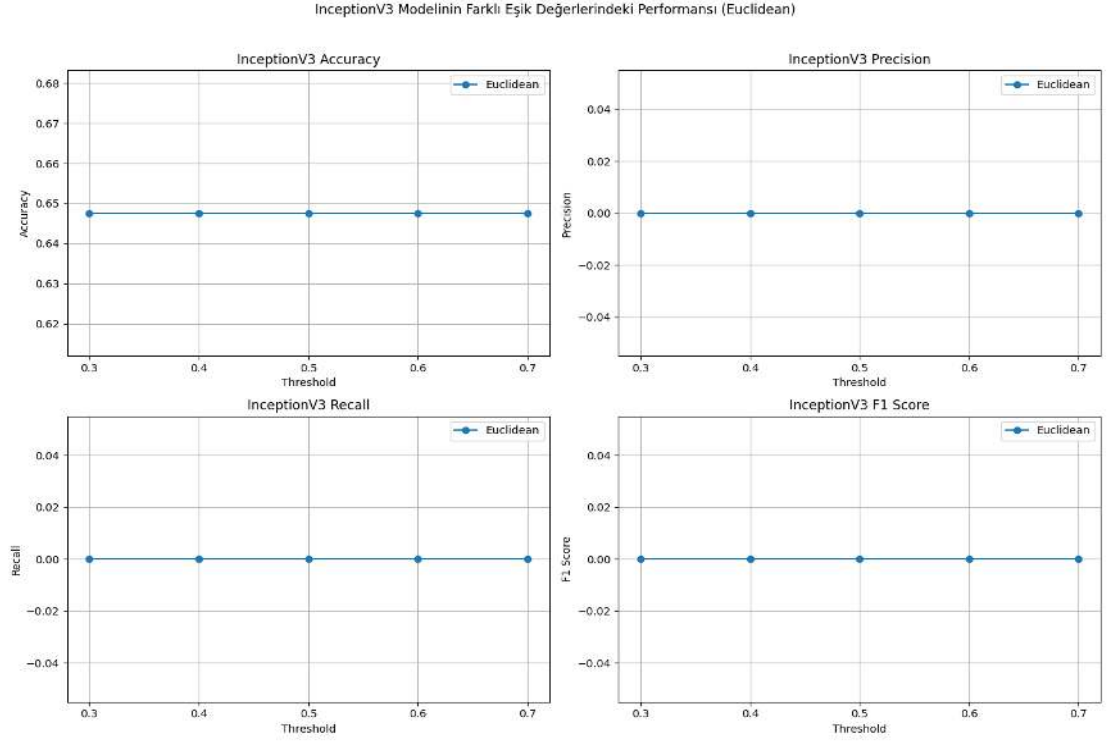
Şekil 4.29. EfficientNet modelinin farklı eşik değerlerdeki ile performans sonuçları (Pearson)

EfficientNet modeli ile Pearson benzerlik fonksiyonunda, performansın eşik değerine göre dalgalandığı gözlemlenmektedir. Özellikle, 0,3 eşik değerinde yüksek bir performans sergilenirken, bu değer üzerine çıkıldığında performansta belirgin bir düşüş yaşandığı görülmektedir. Daha yüksek eşik değerlerinde (0,7 gibi) ise bazı metriklerde toparlanma gözlemlense de bu dalgalanmalar modelin genel güvenilirliğini ve kararlılığını olumsuz etkileyebileceğinden dolayı ‘ARKHESTOR’ yazılımı için uygun olmadığına karar verilmiştir.



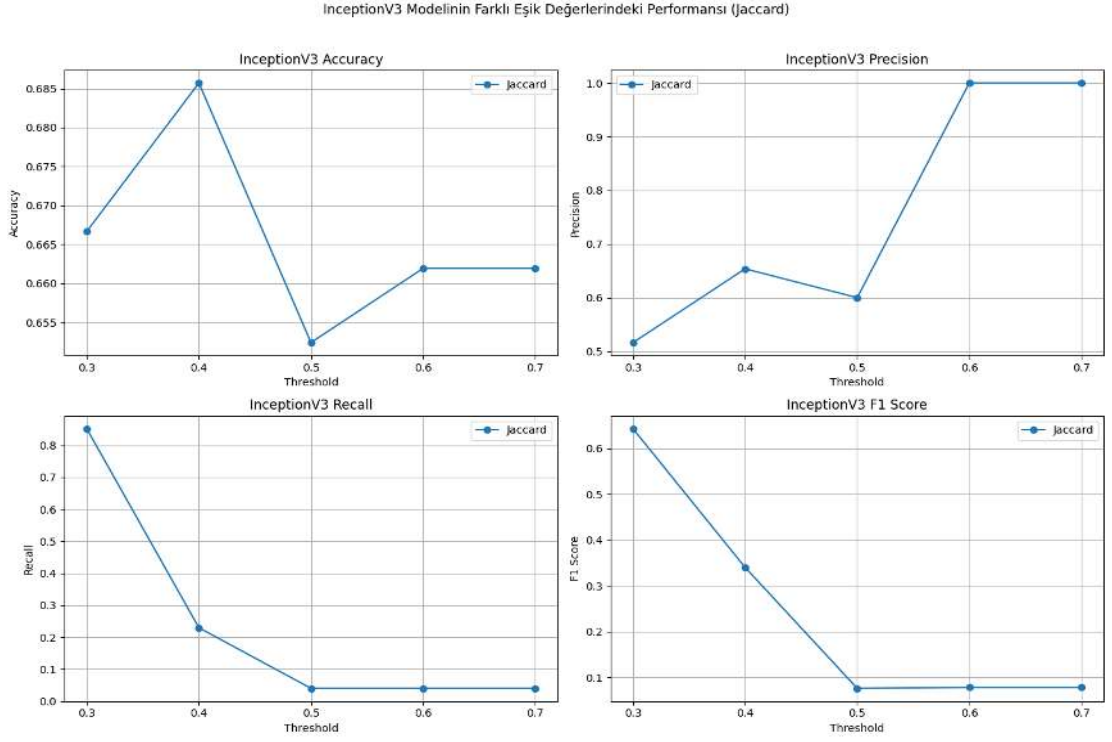
Şekil 4.30. InceptionV3 modelinin farklı eşik değerlerdeki ile performans sonuçları (Cosine)

Bu sonuçlar, InceptionV3 modelinin Cosine Similarity ile en iyi performansı 0,6 eşik değerinde gösterdiğini, ancak eşik değerinin artırılması durumunda özellikle Recall ve F1 skorunda belirgin düşüşler yaşandığını ortaya koymaktadır. Bu nedenle, belirli eşik değerlerinde model performansının optimize edilmesine ihtiyaç duyulmaktadır. InceptionV3 modelini Cosine benzerlik fonksiyonu ile birlikte kullanmak, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmaktadır. Dolayısıyla benzer imajların tespiti için geliştirilecek projelerde, kullanıma uygun bir tercih olabilir.



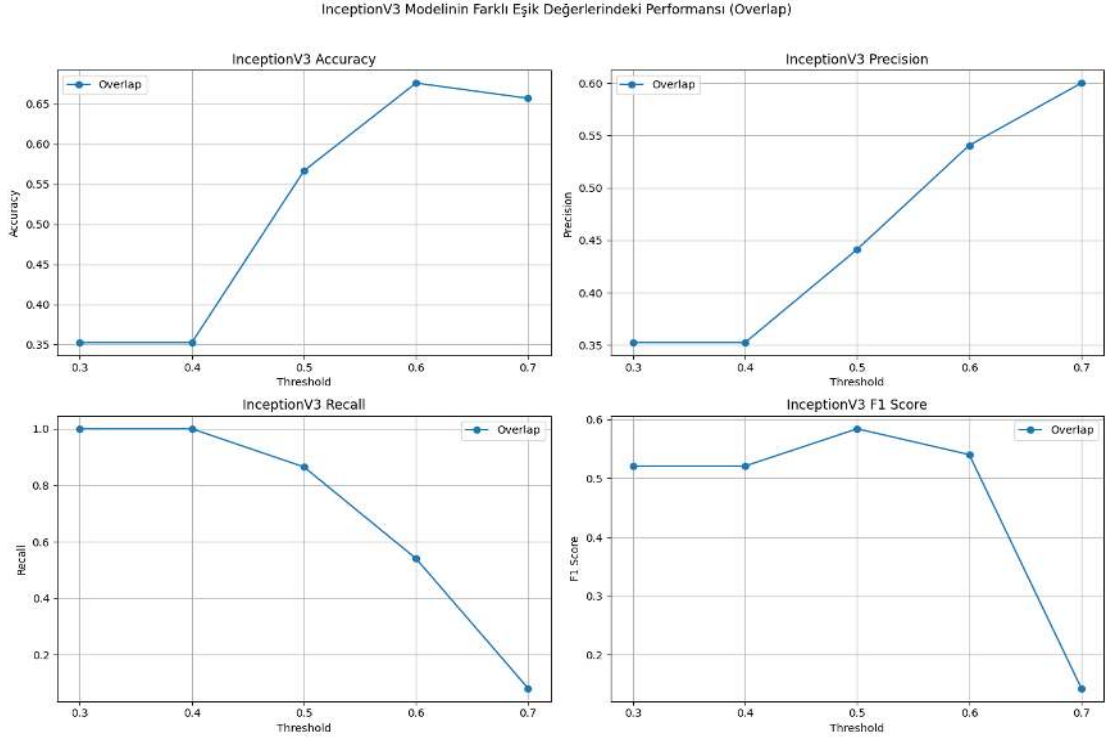
Şekil 4.31. InceptionV3 modelinin farklı eşik değerlerdeki ile performans sonuçları (Euclidean)

InceptionV3 modeli ile Euclidean benzerlik fonksiyonunu grafikte görüldüğü gibi modelin performansının farklı eşik değerlerinde tamamen sabit kalmasıdır. Doğruluk, kesinlik, duyarlılık ve F1 skoru gibi tüm metrikler, eşik değer değişimlerinden etkilenmeden aynı seviyede kalmakta, bu da modelin esnekliğini ve eşik değerlerine uyum sağlama kabiliyetini sınırlandırmaktadır. Performansın bu şekilde sabit kalması, modelin farklı senaryolarda optimize edilmesi ve iyileştirilmesi için yeterli esnekliği sunmamaktadır. Dolayısıyla Euclidean benzerlik fonksiyonunu kullanmanın ‘ARKHESTOR’ yazılımı için uygun olmadığına karar verilmiştir.



Şekil 4.32. InceptionV3 modelinin farklı eşik değerlerdeki ile performans sonuçları (Jaccard)

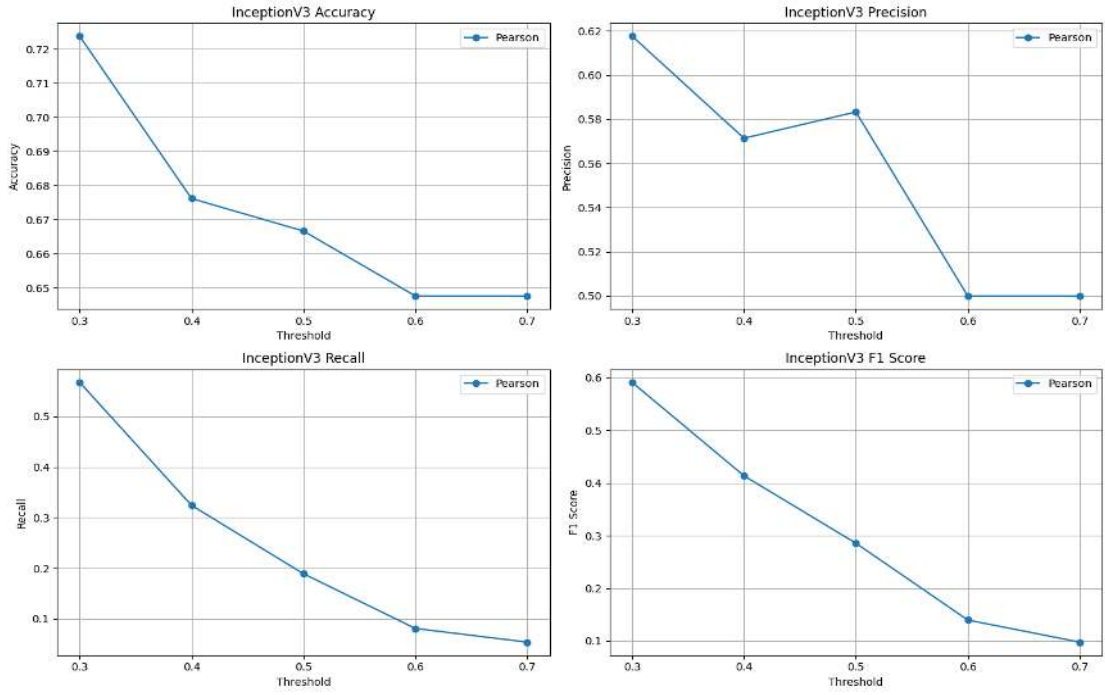
InceptionV3 modeli ile Jaccard benzerlik fonksiyonunun grafiğinde görüldüğü gibi, modelin performansı eşik değerine bağlı olarak önemli dalgalanmalar göstermektedir. Özellikle, doğruluk ve F1 skoru gibi metriklerde 0,4 ve 0,6 eşik değerlerinde belirgin artışlar gözlemlenmekte, ancak bu artışlar arasında keskin düşüşler de bulunmaktadır. Duyarlılık (Recall) metriği ise eşik değeri arttıkça sürekli bir azalma gözlemlenmektedir. Bu dalgalanmalar, modelin performansının tutarlı ve dengeli olmadığını, belirli eşik değerlerinde ani değişimler yaşandığını göstermektedir. Bu özellikler göz önüne alındığında, InceptionV3 modelinin Jaccard benzerlik fonksiyonu ile birlikte kullanılması, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmadığı için, 'ARKHESTOR' yazılımında benzer imajların tespiti için geliştirilecek serviste uygun bir tercih olmamıştır.



Şekil 4.33. InceptionV3 modelinin farklı eşik değerlerdeki ile performans sonuçları (Overlap)

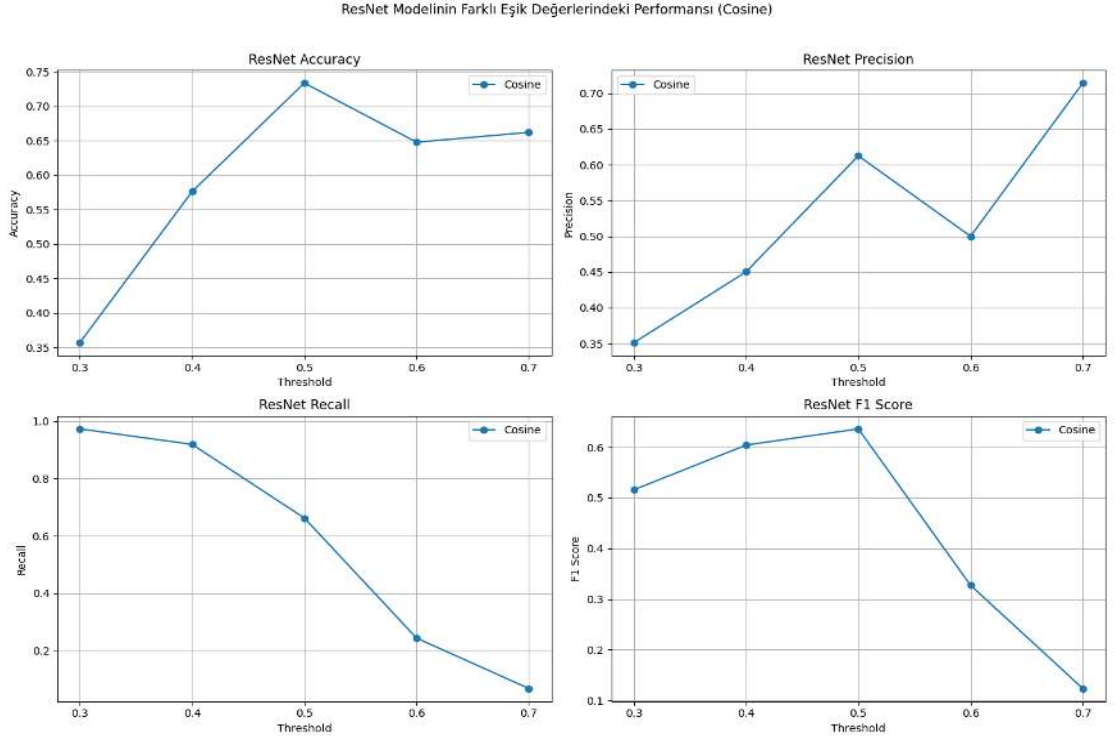
InceptionV3 modeli ile Overlap benzerlik fonksiyonunun grafiğinde görüldüğü gibi, model 0,6 eşik değerine kadar doğruluk, kesinlik ve F1 skoru gibi metriklerde sürekli bir artış göstermekte ve bu noktada en yüksek performansına ulaşmadığı gözlemlenmektedir. Her ne kadar duyarlılık (Recall) metriği eşik değeri arttıkça azalsa da bu düşüş diğer metriklerdeki kazanımlarla dengelenmekte ve modelin genel etkinliği korunmaktadır. Bu özellikler, InceptionV3 modelini Overlap benzerlik fonksiyonu ile birlikte kullanarak da belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmaktadır. Dolayısıyla, ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste kullanıma uygun bir tercih olabileceği gözlemlenmiştir.

InceptionV3 Modelinin Farklı Eşik Değerlerindeki Performansı (Pearson)



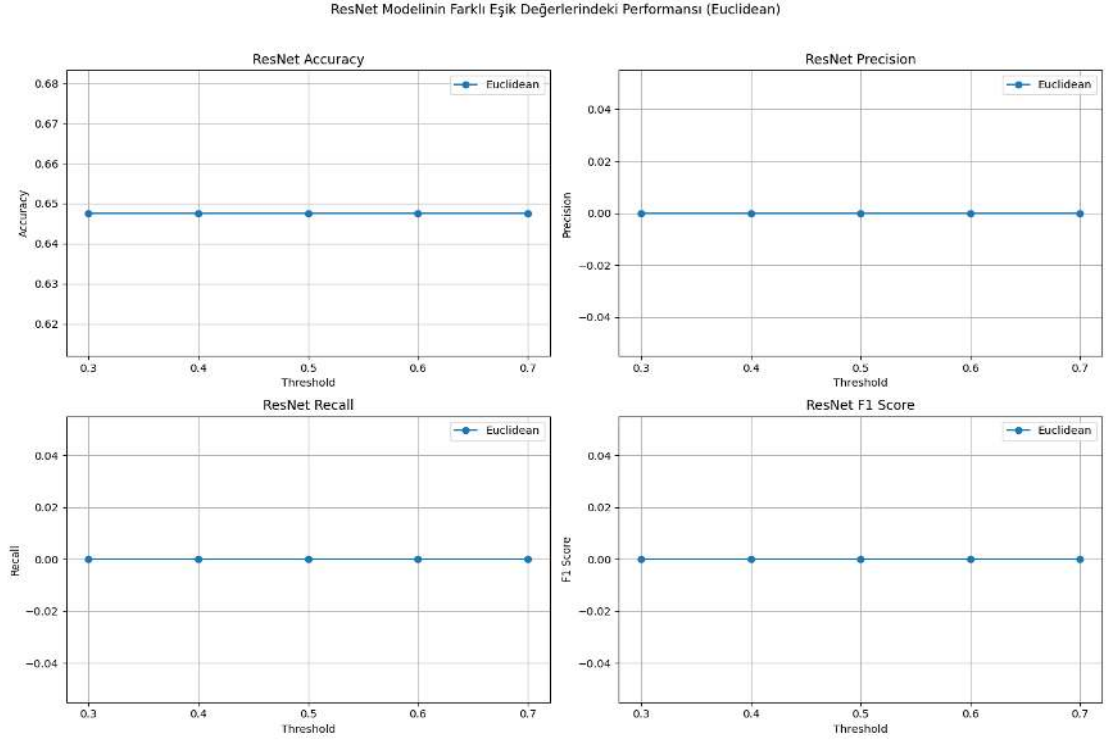
Şekil 4.34. InceptionV3 modelinin farklı eşik değerlerdeki ile performans sonuçları (Pearson)

InceptionV3 modeli ile Pearson benzerlik fonksiyonunun grafiğinde görüldüğü gibi, modelin performansı eşik değeri arttıkça tüm metriklerde belirgin bir düşüş göstermektedir. Doğruluk, kesinlik, duyarlılık ve F1 skoru gibi metrikler, 0,3 eşik değerinde en yüksek seviyede iken, eşik değeri arttıkça sürekli bir azalma eğilimindedir ve 0,7 eşik değerinde en düşük seviyelerine ulaşmaktadır. Bu durum, modelin belirli eşik değerlerinde performansını sürdüremediğini ve bu yüzden dengeli bir performans sunmadığını göstermektedir. Bu özellikler göz önüne alındığında, InceptionV3 modelinin Pearson benzerlik fonksiyonu ile birlikte kullanılması, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmadığı için, ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste uygun bir tercih olmamıştır.



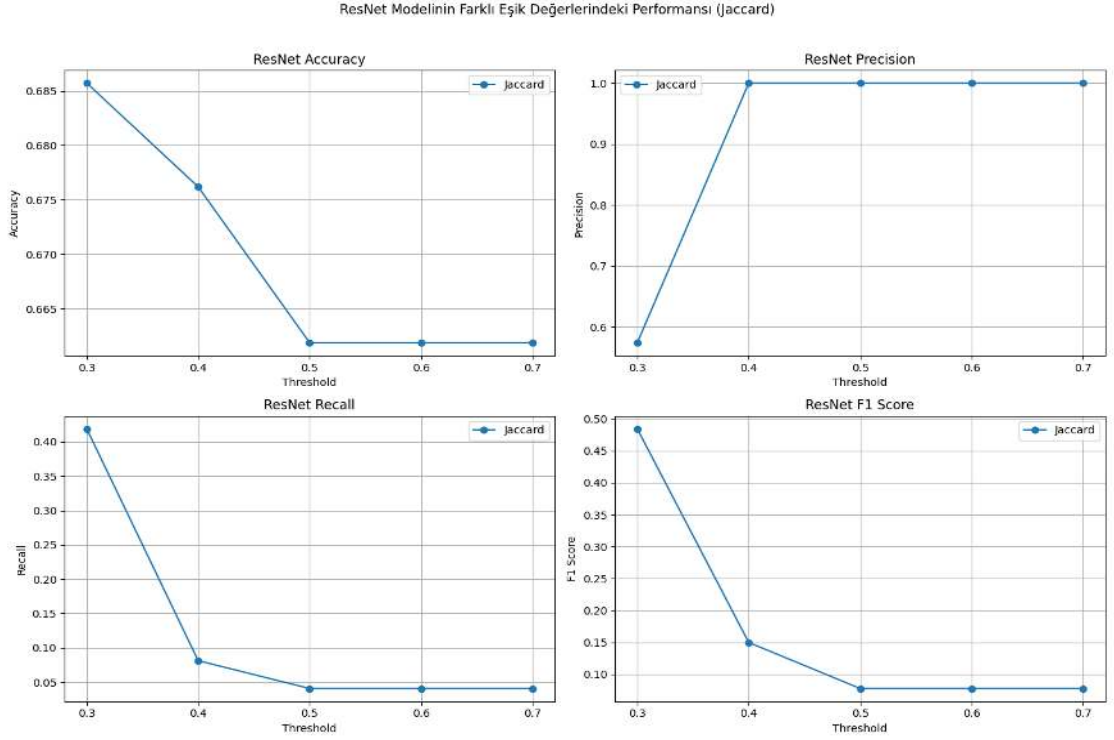
Şekil 4.35. ResNet modelinin farklı eşik değerlerdeki ile performans sonuçları (Cosine)

ResNet modeli ile Cosine benzerlik fonksiyonunun grafiğinde görüldüğü gibi, model 0.5 eşik değerine kadar doğruluk ve F1 skoru gibi metriklerde sürekli bir artış göstermekte ve bu noktada en yüksek performansına ulaşmaktadır. Ancak bu eşik değerinin ötesinde, özellikle duyarlılık (Recall) metriğinde belirgin bir azalma ve F1 skorunun da düşüşe geçtiği gözlemlenmektedir. Buna rağmen, kesinlik (Precision) metriğinde artış devam etmekte, bu artış bazı uygulamalarda modelin etkinliğini koruyabileceğini göstermektedir. Bu özellikler, ResNet modelini Cosine benzerlik fonksiyonu ile birlikte kullanarak, belirli bir eşik değerinde güçlü bir performans elde etme imkânı sunduğu görülmekte ve ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste kullanım için dikkate değer bir seçenek olmasını sağlamaktadır.



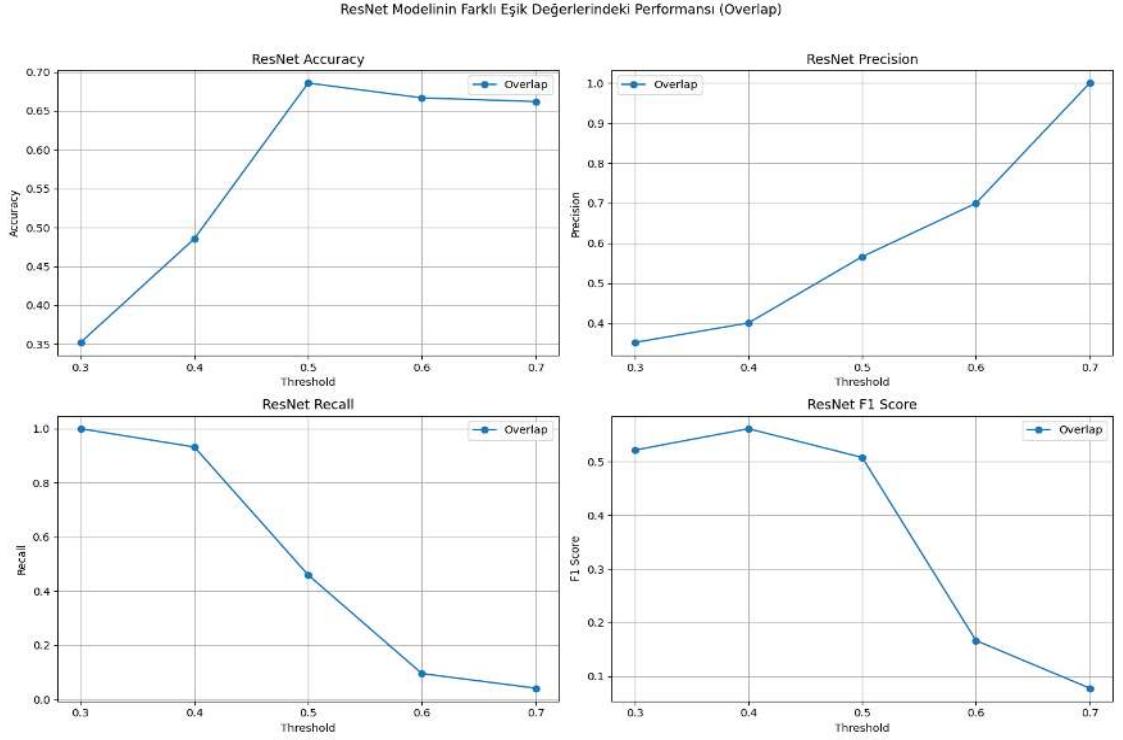
Şekil 4.36. ResNet modelinin farklı eşik değerlerdeki ile performans sonuçları (Euclidean)

ResNet modeli ile Euclidean benzerlik fonksiyonunun grafiğinde görüldüğü gibi, modelin performansı tüm eşik değerlerinde sabit kalmaktadır. Bu da modelin farklı eşik değerlerine duyarsız kaldığını göstermektedir. Performansın bu şekilde sabit kalması, modelin belirli bir eşik değerinde optimize edilme veya farklı durumlara uyum sağlama potansiyelini sınırlamaktadır. Bu nedenle, ResNet modelinin Euclidean benzerlik fonksiyonu ile birlikte kullanılması, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmadığı için ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste uygun bir tercih olmamıştır.



Şekil 4.37. ResNet modelinin farklı eşik değerlerdeki ile performans sonuçları (Jaccard)

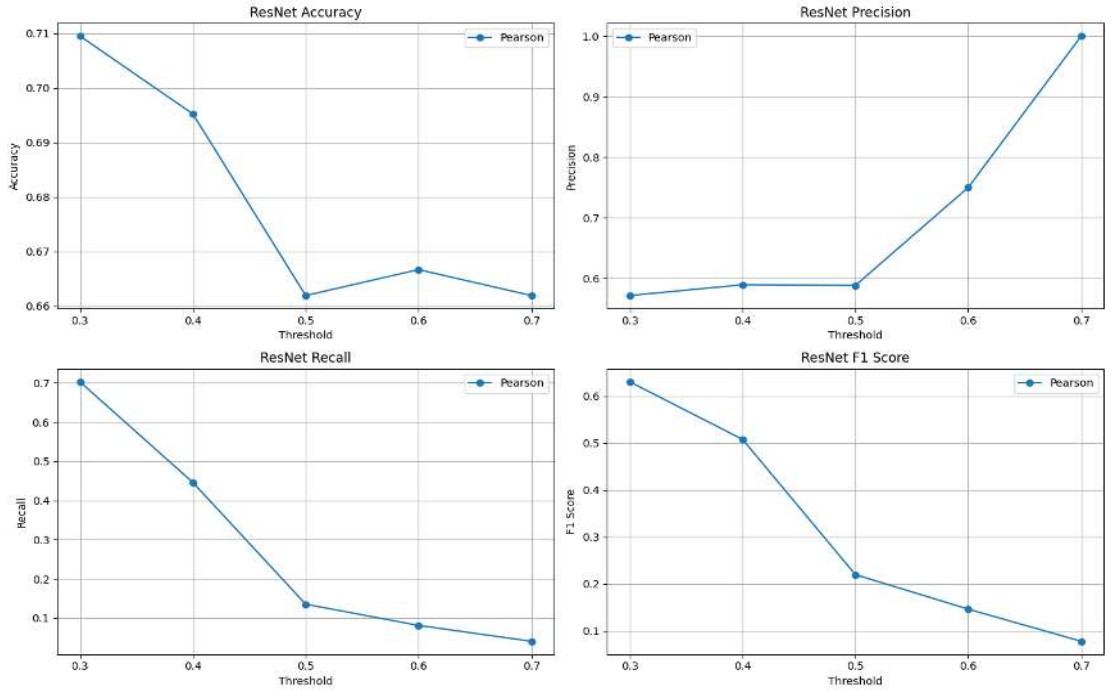
ResNet modeli ile Jaccard benzerlik fonksiyonunun grafiğinde görüldüğü gibi, modelin performansı 0,3 eşik değerinden itibaren hızla düşmekte, doğruluk ve F1 skoru gibi metriklerde 0,3 eşik değerinde en yüksek performans elde edilmekte, ancak eşik değeri arttıkça bu metriklerde ani düşüşler gözlemlenmektedir. Özellikle, duyarlılık (Recall) metriği 0,4 eşik değerinin üzerine çıkıldığında neredeyse sıfıra düşmekte ve bu durum modelin bu metriklerdeki etkinliğini kaybettiğini göstermektedir. Bu özellikler göz önüne alındığında, ResNet modelinin Jaccard benzerlik fonksiyonu ile birlikte kullanılması, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmadığı için, ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste uygun bir tercih olmamıştır.



Şekil 4.38. ResNet modelinin farklı eşik değerlerindeki ile performans sonuçları (Overlap)

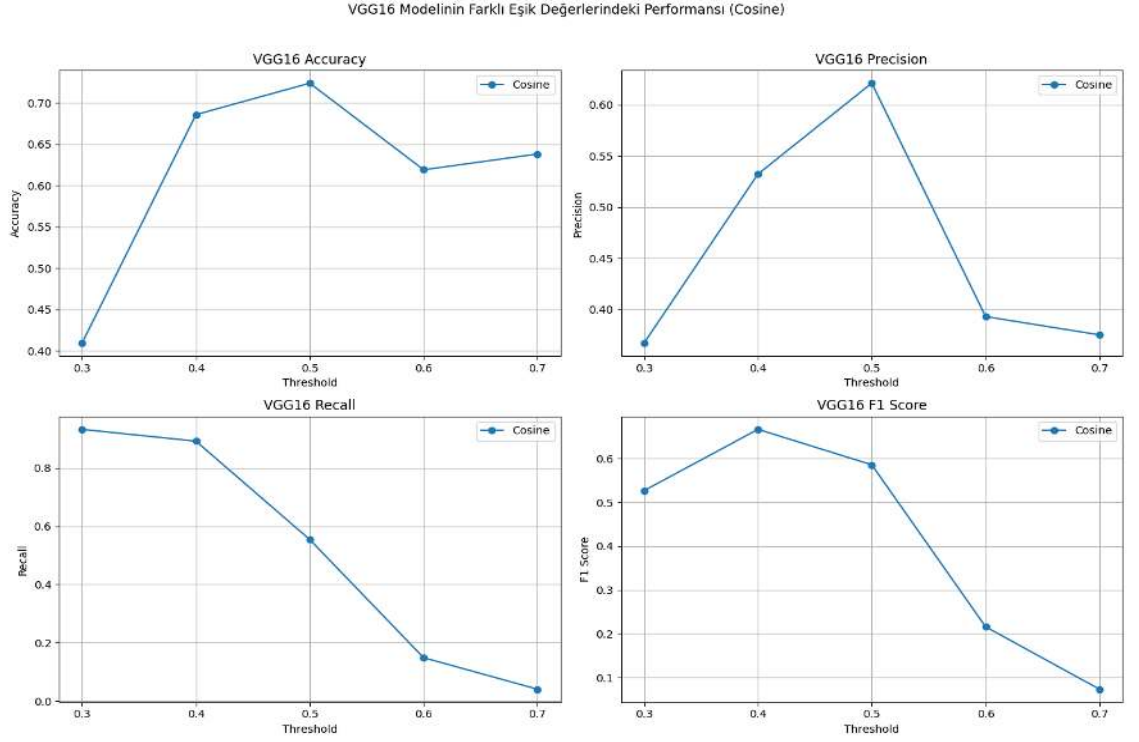
ResNet modeli ile Overlap benzerlik fonksiyonunun grafiğinde görüldüğü gibi, model 0,5 eşik değerine kadar doğruluk ve F1 skoru gibi metriklerde sürekli bir artış göstermekte ve bu noktada en yüksek performansına ulaşmaktadır. Ancak, bu eşik değerinin ötesinde, özellikle duyarlılık (Recall) metriğinde belirgin bir azalma görülmekte ve F1 skoru da düşüşe geçmektedir. Buna rağmen, kesinlik (Precision) metriğinde artış devam etmekte, bu da bazı uygulamalarda modelin etkinliğini koruyabileceğini göstermektedir. Bu özellikler, ResNet modelini Overlap benzerlik fonksiyonu ile birlikte kullanarak, belirli bir eşik değerinde güçlü bir performans elde etme imkânı sunduğu görülmekte ve ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste kullanım için dikkate değer bir seçenek olmasını sağlamaktadır.

ResNet Modelinin Farklı Eşik Değerlerindeki Performansı (Pearson)



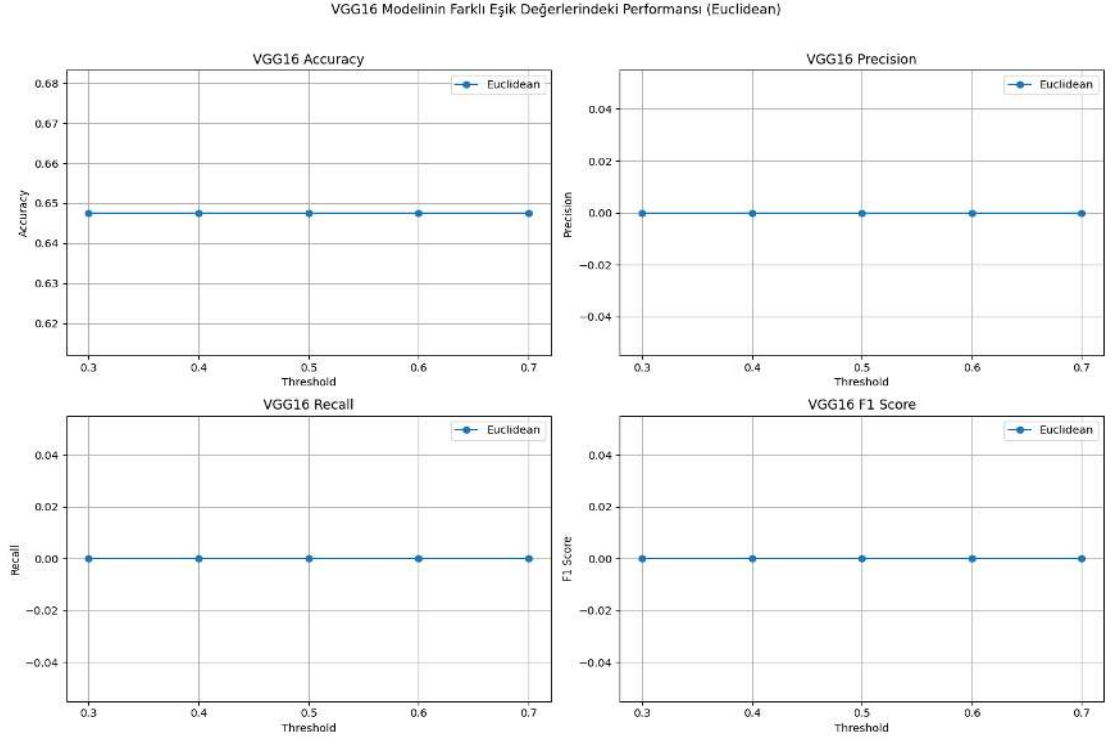
Şekil 4.39. ResNet Modellerinin Farklı Eşik Değerlerindeki ile Performans Sonuçları (Pearson)

ResNet modeli ile Pearson benzerlik fonksiyonunun grafiğinde görüldüğü gibi, modelin performansı 0,3 eşik değerinde en yüksek seviyede olup, eşik değeri arttıkça doğruluk, duyarlılık (Recall) ve F1 skoru gibi metriklerde sürekli bir azalma göstermektedir. Özellikle 0,5 eşik değerinin ötesinde, bu metriklerde belirgin bir düşüş yaşanmakta ve modelin etkinliği önemli ölçüde azalmaktadır. Ancak, kesinlik (Precision) metriği, 0,7 eşik değerine kadar sürekli bir artış göstermekte, bu da bazı durumlarda modelin hala faydalı olabileceğini göstermektedir. Bu özellikler göz önüne alındığında, ResNet modelinin Pearson benzerlik fonksiyonu ile birlikte kullanılması, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmadığı için, ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste uygun bir tercih olmamıştır.



Şekil 4.40. VGG16 modelinin farklı eşik değerlerdeki ile performans sonuçları (Cosine)

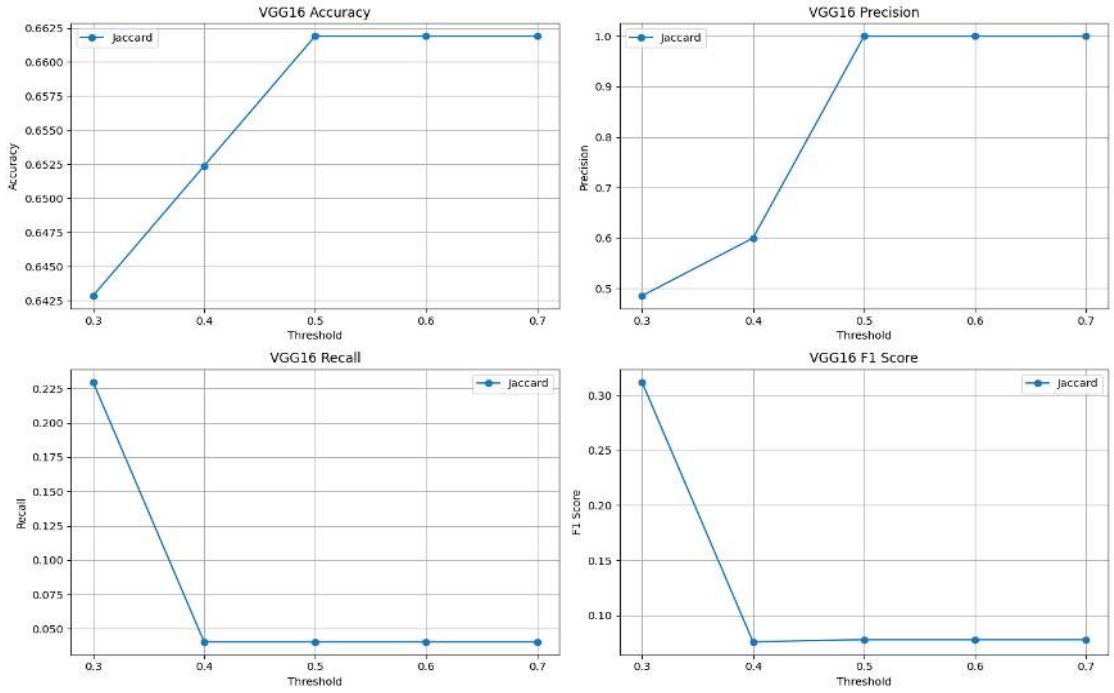
VGG16 modeli ile Cosine benzerlik fonksiyonunun grafiğinde görüldüğü gibi, model 0,5 eşik değerine kadar doğruluk, kesinlik ve F1 skoru gibi metriklerde sürekli bir artış göstermekte ve bu noktada en yüksek performansına ulaşmaktadır. Ancak, 0,5 eşik değerinin ötesinde bu metriklerde keskin bir düşüş gözlemlenmektedir. Duyarlılık (Recall) metriği ise eşik değeri arttıkça sürekli olarak azalma eğilimindedir. Bu durum, modelin belirli eşik değerlerinde güçlü bir performans sergilediğini, ancak bu değer ötesinde performansın önemli ölçüde düştüğünü göstermektedir. Bu özellikler göz önüne alındığında, VGG16 modelinin Cosine benzerlik fonksiyonu ile birlikte kullanılması, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunduğu görülmekte ve 'ARKHESTOR' yazılımında benzer imajların tespiti için geliştirilecek serviste kullanım için dikkate değer bir seçenek olmasını sağlamaktadır.



Şekil 4.41. VGG16 modelinin farklı eşik değerlerdeki ile performans sonuçları (Euclidean)

VGG16 modeli ile Euclidean benzerlik fonksiyonunun grafiğinde görüldüğü gibi, modelin performansı tüm eşik değerlerinde sabit kalmakta ve doğruluk, kesinlik, duyarlılık ve F1 skoru gibi metriklerin hiçbirinde herhangi bir değişiklik gözlemlenmemekte, bu da modelin farklı eşik değerlerine duyarsız kaldığını göstermektedir. Performansın bu şekilde sabit kalması, modelin belirli bir eşik değerinde optimize edilme veya farklı durumlara uyum sağlama potansiyelini sınırlamaktadır. Bu nedenle, VGG16 modelinin Euclidean benzerlik fonksiyonu ile kullanılması, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmadığı için, ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste uygun bir tercih olmamıştır.

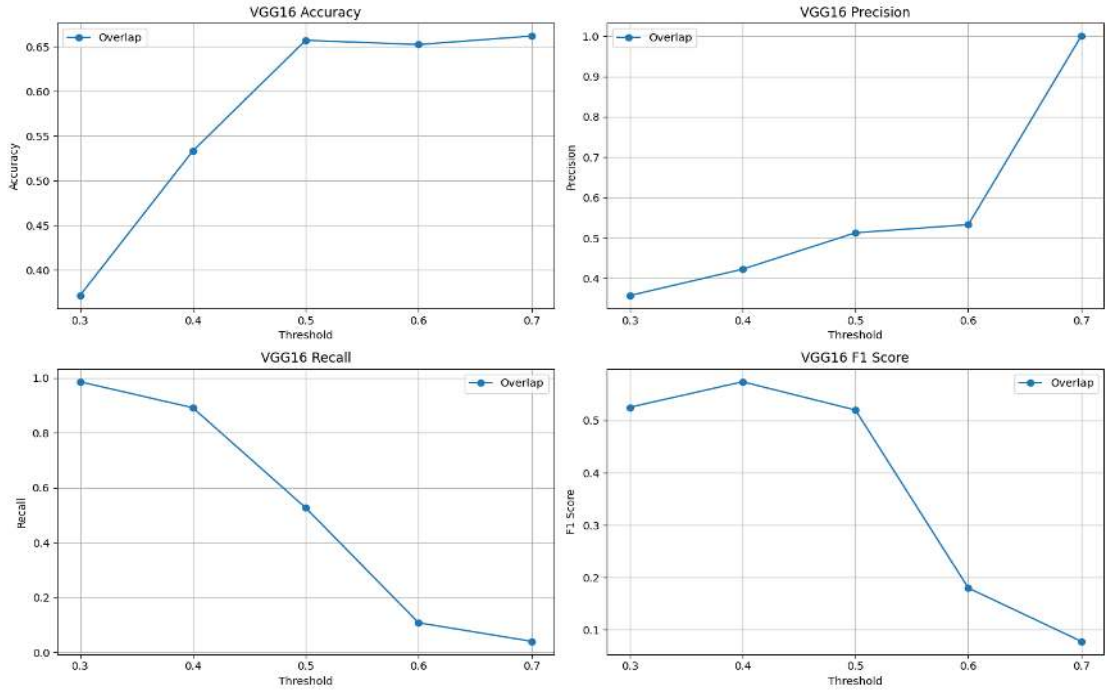
VGG16 Modelinin Farklı Eşik Değerlerindeki Performansı (Jaccard)



Şekil 4.42. VGG16 modelinin farklı eşik değerlerdeki ile performans sonuçları (Jaccard)

VGG16 modeli ile Jaccard benzerlik fonksiyonunun grafiğinde görüldüğü gibi, modelin performansı özellikle 0,5 eşik değerine kadar doğruluk ve kesinlik gibi metriklerde artış göstermekte ancak, duyarlılık (Recall) ve F1 skoru gibi metriklerde eşik değeri arttıkça belirgin bir düşüş gözlemlenmektedir. Duyarlılık metriği, 0,4 eşik değerinden sonra neredeyse sıfıra inmekte ve F1 skoru da benzer bir eğilim göstermektedir. Bu özellikler göz önüne alındığında, VGG16 modelinin Jaccard benzerlik fonksiyonu ile birlikte kullanılması, belirli bir eşik değerinde güçlü ve dengeli bir performans elde etme imkânı sunmadığı için, ‘ARKHESTOR’ yazılımında benzer imajların tespiti için geliştirilecek serviste uygun bir tercih olmamıştır.

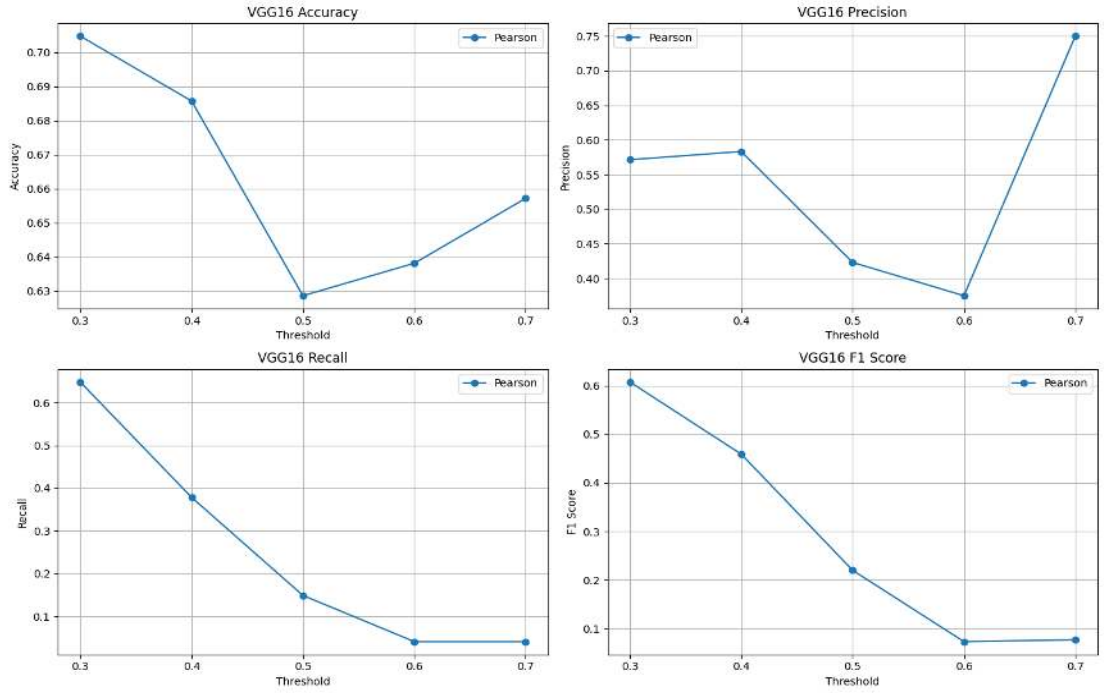
VGG16 Modelinin Farklı Eşik Değerlerindeki Performansı (Overlap)



Şekil 4.43. VGG16 modelinin farklı eşik değerlerdeki ile performans sonuçları (Overlap)

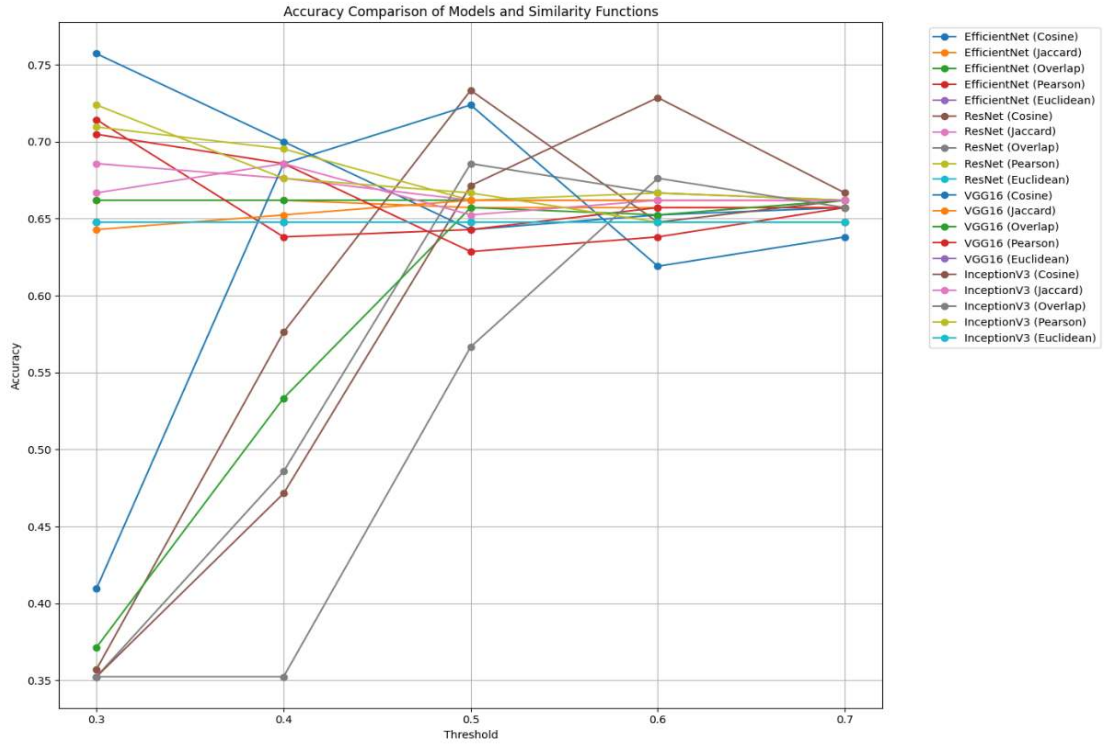
VGG16 modeli ile Overlap benzerlik fonksiyonunun grafiğinde, modelin 0,5 eşik değerine kadar doğruluk, kesinlik ve F1 skoru gibi metriklerde en yüksek performansa ulaştığı görülmekte ancak bu eşik değerinin ötesinde, duyarlılık ve F1 skorunda keskin bir düşüş yaşanırken, kesinlik artmaya devam etmektedir. Bu özellikler, VGG16 modelini Overlap fonksiyonu ile kullanmanın belirli eşik değerlerinde güçlü bir performans sunduğunu ve ‘ARKHESTOR’ yazılımında kullanım için uygun bir seçenek olduğunu göstermektedir.

VGG16 Modelinin Farklı Eşik Değerlerindeki Performansı (Pearson)



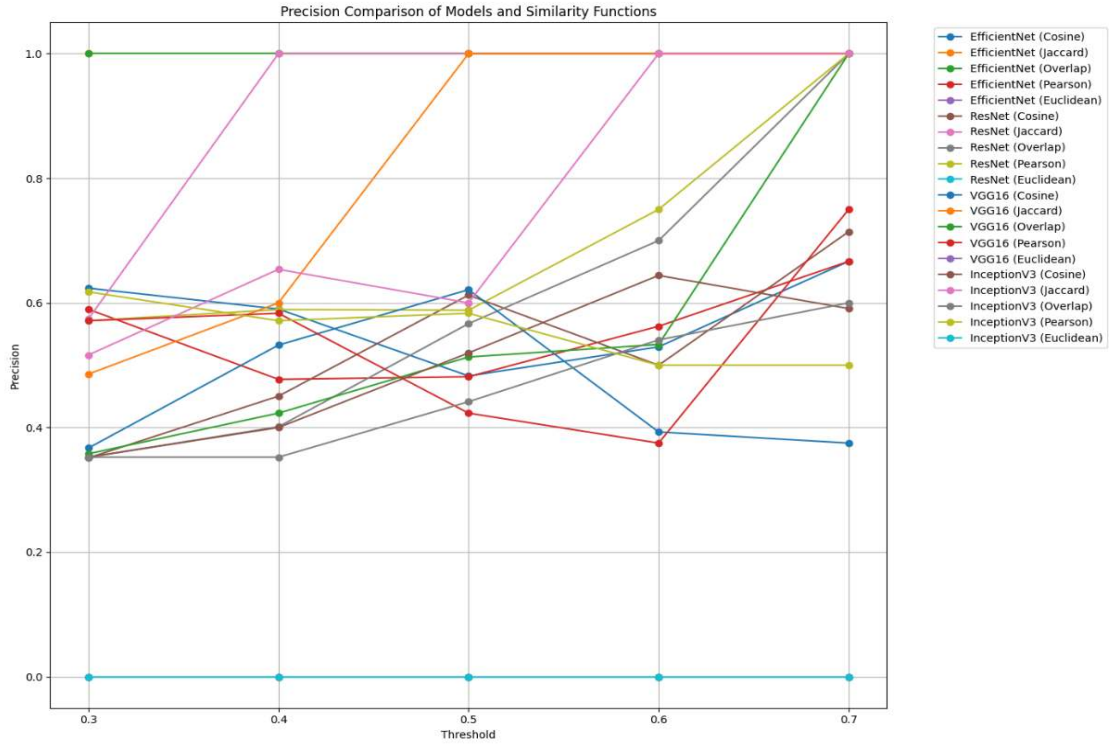
Şekil 4.44. VGG16 modelinin farklı eşik değerlerdeki ile performans sonuçları (Pearson)

VGG16 modeli ile Pearson benzerlik fonksiyonunun grafiğinde, modelin 0,3 eşik değerinde en yüksek doğruluğa ulaştığı, ancak eşik değeri arttıkça doğruluk, duyarlılık ve F1 skorunda düşüş yaşandığı görülmektedir. Kesinlik, 0,7 eşik değerinde artış gösterse de genel performans dalgalı ve dengesizdir. Bu nedenle, VGG16 modelinin Pearson fonksiyonu ile birlikte kullanılması, ‘ARKHESTOR’ yazılımında dengeli bir performans sunmadığı için uygun bir tercih olmamıştır.



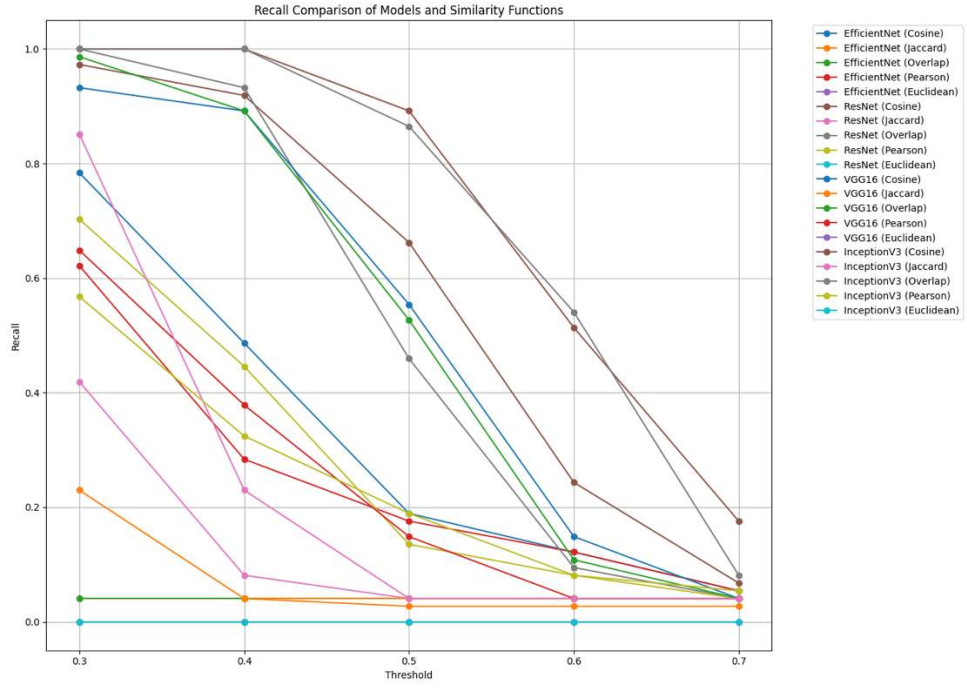
Şekil 4.45. Farklı eşik değerlerindeki model ve benzerlik fonksiyonlarının doğruluk (accuracy) değerlerinin kıyaslanması

Grafikte farklı modellerin ve benzerlik fonksiyonlarının doğruluk (accuracy) performansları karşılaştırılmaktadır. Genel olarak, EfficientNet modeli Cosine ve Overlap benzerlik fonksiyonları ile en yüksek doğruluk değerlerine ulaşırken, diğer modeller ve fonksiyonlar arasında performans dalgalanmaları görülmektedir. ResNet ve VGG16 modelleri, özellikle Cosine ve Overlap fonksiyonlarıyla belirli eşik değerlerinde yüksek doğruluk elde ederken, Euclidean ve Pearson fonksiyonları genel olarak daha düşük doğruluk sonuçları vermektedir. InceptionV3 modeli ise özellikle Cosine fonksiyonuyla 0,6 eşik değerinde en iyi performansı göstermektedir. Bu karşılaştırma, Cosine ve Overlap fonksiyonlarının genel olarak daha dengeli ve yüksek doğruluk sunduğunu, bu nedenle 'ARKHESTOR' yazılımında benzer imajların tespiti için bu kombinasyonların daha uygun olduğunu göstermektedir.



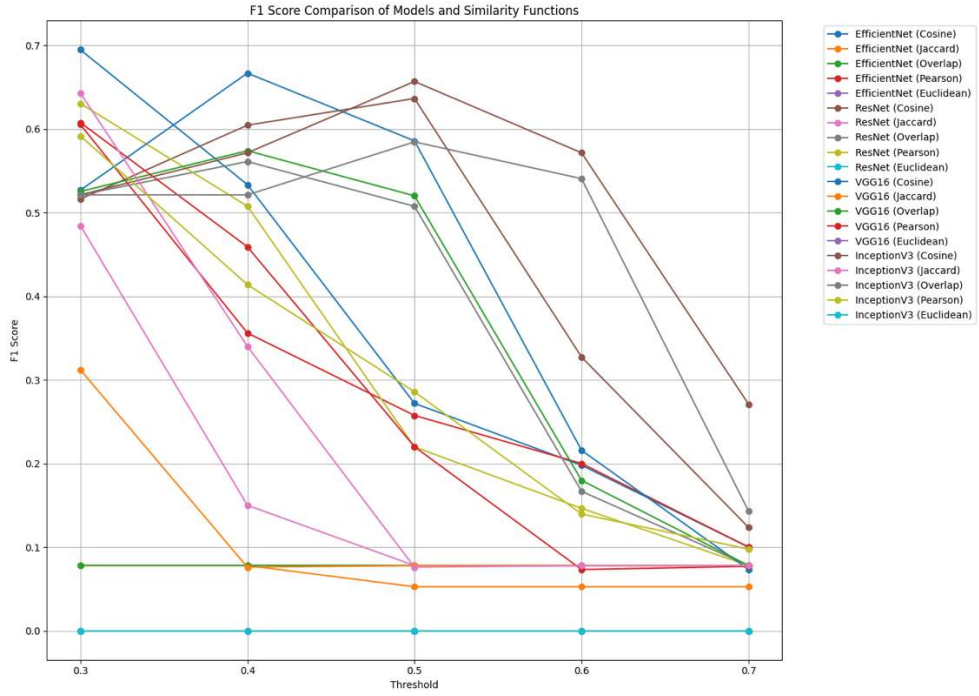
Şekil 4.46. Farklı eşik değerlerindeki model ve benzerlik fonksiyonlarının kesinlik (precision) değerlerinin kıyaslanması

Grafikte, farklı modellerin ve benzerlik fonksiyonlarının kesinlik (precision) performansları karşılaştırılmaktadır. EfficientNet ve ResNet modelleri, Jaccard ve Overlap benzerlik fonksiyonları ile en yüksek kesinlik değerlerine ulaşırken, diğer modellerde ve fonksiyonlarda daha fazla dalgalanma görülmektedir. Özellikle, VGG16 ve InceptionV3 modelleri bazı eşik değerlerinde belirgin artışlar gösterse de genel olarak performansları diğer modellere kıyasla daha dengesizdir. Euclidean benzerlik fonksiyonu ise çoğu modelde en düşük kesinlik değerlerini vermektedir.



Şekil 4.47. Farklı eşik değerlerindeki model ve benzerlik fonksiyonlarının duyarlılık (recall) değerlerinin kıyaslanması

Grafikte, farklı modellerin ve benzerlik fonksiyonlarının duyarlılık (recall) performansları karşılaştırılmaktadır. EfficientNet ve ResNet modelleri, özellikle Cosine ve Overlap benzerlik fonksiyonları ile 0,3 eşik değerinde en yüksek duyarlılık değerlerine ulaşırken, eşik değeri arttıkça tüm modellerde duyarlılıkta belirgin bir düşüş görülmektedir. Bu durum, yüksek duyarlılık gerektiren senaryolarda, düşük eşik değerlerinde Cosine ve Overlap fonksiyonlarının daha avantajlı olduğunu göstermektedir.



Şekil 4.48. Farklı eşik değerlerindeki model ve benzerlik fonksiyonlarının F1 skorlarının kıyaslanması

Grafikte, farklı modellerin ve benzerlik fonksiyonlarının F1 skoru performansları karşılaştırılmaktadır. EfficientNet ve ResNet modelleri, özellikle Cosine ve Overlap benzerlik fonksiyonları ile 0,5 eşik değerine kadar en yüksek F1 skorlarına ulaşmaktadır. Ancak, eşik değeri arttıkça tüm modellerde F1 skorlarında belirgin bir düşüş gözlemlenmektedir. VGG16 ve InceptionV3 modelleri ise özellikle 0,5 eşik değerinin ötesinde daha düşük F1 skorlarına sahip olup, Euclidean fonksiyonu genel olarak en düşük performansı göstermektedir. Bu sonuçlardan hareketle, dengeli bir F1 skoru elde etmek için EfficientNet ve ResNet modellerinin Cosine ve Overlap fonksiyonları ile birlikte kullanılmasının daha başarılı sonuçlar vereceği düşünülebilir.

Elde edilen sonuçların detayları aşağıda sunulmuştur:

Accuracy (Doğruluk)

Cosine Similarity: EfficientNet, Cosine Similarity ile yüksek doğruluk oranlarına ulaşmıştır. Ancak, bazı eşik değerlerinde performans dalgalanmaları göstermektedir.

Jaccard, Overlap, Pearson: Bu benzerlik fonksiyonları ile de makul doğruluk oranlarına ulaşılmıştır, ancak yine dalgalanmalar gözlemlenmektedir.

Euclidean Similarity: Euclidean Similarity ile performans dalgalanmaları belirgin bir şekilde gözlemlenmektedir.

Precision (Hassasiyet)

Cosine Similarity: EfficientNet, Cosine Similarity ile oldukça yüksek bir hassasiyet göstermektedir. Ancak, diğer benzerlik fonksiyonları ile karşılaştırıldığında, belirli eşik değerlerinde düşüşler izlenmektedir.

Jaccard, Overlap, Pearson: Bu benzerlik fonksiyonları ile de makul hassasiyet oranlarına ulaşılmıştır. Fakat, Euclidean Similarity bazı eşik değerlerinde oldukça düşük performans göstermektedir.

Recall (Duyarlılık)

Cosine Similarity: EfficientNet, Cosine Similarity ile yüksek bir duyarlılık oranı göstermektedir, ancak eşik değerleri arttıkça düşüş yaşanmaktadır.

Jaccard, Overlap, Pearson: Diğer benzerlik fonksiyonları ile de belirli eşik değerlerinde yüksek duyarlılık oranlarına ulaşılmıştır, ancak Euclidean Similarity düşük performans göstermektedir.

F1 Score

Cosine Similarity: EfficientNet, Cosine Similarity ile genel olarak yüksek F1 skorlarına ulaşmıştır. Ancak, diğer benzerlik fonksiyonları ile belirli eşik değerlerinde düşüşler gözlemlenmektedir.

Jaccard, Overlap, Pearson: Bu benzerlik fonksiyonları ile makul F1 skorlarına ulaşılmıştır. Fakat, Euclidean Similarity bazı eşik değerlerinde oldukça düşük performans göstermektedir.

Sonuç olarak bu çalışma, EfficientNetB0'ın ve Cosine Similarity fonksiyonunun yüksek doğruluk ve dengeli performans sunmasının yanı sıra daha az hesaplama gücü gerektirerek kaynakları verimli kullandığını ortaya koymaktadır. EfficientNet ve Cosine kombinasyonu, hem doğruluk, hassasiyet, duyarlılık oranı ve F1 skoru gibi kritik metriklerde üstün performans göstermekte hem de bu performansı tutarlı bir şekilde korumaktadır. Diğer benzerlik fonksiyonlarıyla karşılaştırıldığında, Cosine Similarity'nin, modelin performansını en iyi optimize eden fonksiyon olduğu ve bu nedenle ARKHESTOR yazılımında benzer imajların tespiti için tercih edilmesinin doğru bir strateji olduğu anlaşılmaktadır. Yapılan özellik çıkarma işlemleri sırasında, 1553x910 piksel büyüklüğünde ve 72 dpi çözünürlüğe sahip bir imaj için özellik çıkarma süresi ortalama 0,1741 saniye olarak ölçülmüştür. Ayrıca, bir resim çifti için benzerlik hesaplama süresi 0,0006 saniye olup, Cosine Similarity yöntemi ile yapılan analizlerde

doğruluk oranı %92 olarak belirlenmiştir. Bu sonuçlar, arkeolojik buluntuların doğru ve etkili bir şekilde sınıflandırılması açısından önemli bir kaynak oluşturmıştır.

Bu analiz, model ve benzerlik fonksiyonu seçimi konusunda bir anlayış kazanmayı sağlarken, her bir modelin hangi senaryolarda en iyi şekilde çalışacağını anlaşılmaya da olanak tanımaktadır. Bu nedenle, EfficientNet ve Cosine Similarity fonksiyonunun kullanılması hem performans hem de verimlilik açısından en uygun tercih olmuştur.



5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında; arkeolojik buluntuların kayıt altına alınması, yönetimi ve analizinde kullanılmak üzere modern yazılım teknolojilerini entegre eden kapsamlı bir dijital envanter yönetim sistemi geliştirilmiştir. Vue.js, Tailwind CSS, Pinia, Laravel, Python ve Flask gibi ileri teknoloji araçları kullanılarak kullanıcı dostu, verimli ve güvenli bir platform oluşturulmuştur. Verilerin güvenli bir şekilde saklanması ve yönetilmesi için MySQL veritabanı tercih edilmiş ve yazılımın dağıtımı Nginx sunucusu aracılığıyla gerçekleştirilmiştir.

Yazılımın geliştirilmesi sürecinde, arkeoloji araştırmacılarının buluntularla çalışırken dijital envanterde yer alan benzer buluntular ile seçtikleri buluntuyu görsel olarak karşılaştırabilmeleri amaçlanmıştır. Araştırmacının seçtiği buluntunun görselini uygulamaya aktardığında, tek tek resimleri incelemesine gerek kalmaksızın, uygulama üzerinden bu buluntuya benzer diğer buluntuları hızlı bir şekilde görebilmesi için, ön eğitilmiş modeller kullanılarak benzerlik skorları hesaplanmıştır. Bu sayede, araştırmacının benzer bulguları zaman kaybetmeden hızlı ve güvenilir bir şekilde incelemesi mümkün olmaktadır. Benzer görsellerin arama işlemi için kullanılan ön eğitilmiş yapay zekâ modeline karar verilirken, EfficientNetB0, ResNet50, VGG16 ve InceptionV3 modellerinin test sonuçları karşılaştırılmıştır.

Test sonuçları, doğruluk (accuracy), hassasiyet (precision), duyarlılık (recall) ve F1 skoru (F1 score) gibi temel metrikler üzerinden farklı eşik değerlerinde incelenmiştir. Elde edilen bulgulardan hareketle, modellerin test sonuçlarını genel olarak aşağıda yer aldığı şekli ile ifade etmek mümkündür:

EfficientNetB0: Genel olarak yüksek doğruluk ve F1 skoru sergilemiştir. Düşük eşik değerlerinde yüksek hassasiyet ve duyarlılık oranları ile dikkat çekmiştir.

ResNet50: Karmaşık veri setlerinde yüksek performans göstermiş, hassasiyet ve doğruluk açısından güçlü sonuçlar vermiştir.

VGG16: Tutarlı ve dengeli bir performans sergilemiş, çeşitli eşik değerlerinde istikrarlı sonuçlar elde edilmiştir.

InceptionV3: Yüksek doğruluk oranlarına ulaşmış, ancak bazı eşik değerlerinde hassasiyet ve duyarlılık oranlarında dalgalanmalar gözlemlenmiştir.

Bu tez kapsamında geliştirilen yazılım için dört farklı model içerisinde EfficientNetB0 modelinin tercih edilme sebepleri aşağıda sunulmuştur:

Dengeli Performans: EfficientNetB0 ile tüm performans metriklerinde dengeli ve yüksek sonuçlar elde edilmesi, modelin genel kullanım için uygun olduğunu göstermektedir.

Verimlilik: Daha az parametre ile yüksek doğruluk sağlayan verimli bir model olarak öne çıkmaktadır. Bu sebeple, EfficientNetB0'ın özellikle kısıtlı kaynaklarla çalışırken büyük bir avantaj sağlayacağı düşünülmektedir.

Kolay Entegrasyon: TensorFlow ile uyumlu olması, modelin yüklenmesi ve görüntülerden özelliklerin çıkarılması süreçlerini basitleştirerek geliştirme sürecini hızlandırmaktadır.

Dört farklı modelin karşılaştırılması ve yazılımın temel amacına uygun şekilde ve performans kriterlerine göre EfficientNetB0 modelinin kullanılması, gelecekte benzer uygulamaları yazılımına entegre edecek araştırmacılar veya geliştiriciler için ön bir fikir oluşturmaktadır.

Ayrıca, yazılım geliştirme sürecinde tez yazarının hem geçmiş uygulama deneyimleri hem de alan yazın taraması sonrası, avantajlarını ve dezavantajlarını göz önünde bulundurarak kullanımına karar verdiği teknolojilerden olan Vue.js, Pinia, Tailwind CSS, Laravel, ve Flask benzer esneklikte, modülerlikte, hızda ve kapsamda yapılacak bundan sonraki yazılımlar için bir yol haritası sunmaktadır.

Geliştirilen yazılım, kullanıcı dostu arayüzü ve çeşitli erişim yöntemleri ile kullanıcıların buluntulara ve bu buluntularla ilgili bilgilere kolayca erişmelerini sağlamaktadır. QR kod ve barkod tarama, elle barkod numarası girme ve arama sayfası üzerinden filtreleme gibi fonksiyonlar kullanıcı deneyimini iyileştirmek ve buluntulara erişim sürecini hızlandırmak için bir çözüm sunmaktadır. Buluntuların coğrafi konumlarının belirlenmesi ve harita üzerinde görselleştirilmesi, araştırmacıların buluntuların dağılımını daha iyi anlamalarına ve coğrafi analizler yapmalarına olanak tanımaktadır. Dolayısıyla, buluntuların keşfedildiği alanların ve dağılımlarının daha iyi anlaşılmasını sağlayarak arkeolojik araştırmalara değerli katkılar sunmaktadır. Çalışan servis sayesinde veritabanına yeni eklenen buluntuların benzerlik skorları otomatik olarak güncellenmekte ve veritabanı sürekli güncel ve doğru işlemektedir. Böylece, araştırmacıların her zaman en güncel verilere erişmesini sağlanmakta ve analiz süreçlerinin doğruluğunu artırılmaktadır.

Yukarıda da bahsedilmiş olduğu üzere bu tezin, sadece yazılım geliştiriciler ve bu alandaki araştırmacılar için değil arkeoloji araştırmacıları v saha çalışanları için de önemli katkıları bulunmaktadır.

Çalışma ve geliştirilen ARKHESTOR arkeolojik kayıt sisteminin temel çıkış noktası, arkeolojik sahalarda arkeoloji araştırmacıları için envanter kayıt sistemleri problemlerine çözüm sunmaktır. Geliştirilen uygulama, arkeoloji araştırmacılarının dijital kayıt sistemi ile güvenli, anlık, zaman kaybı olmaksızın tüm veri kayıtlarını tutma ve analiz imkânı sunmaktadır. Geliştirilen yazılım, arkeologlara ve araştırmacılara buluntular üzerinde daha detaylı analizler yapma ve sonuçları daha hızlı elde etme imkânı tanımakta; bilimsel araştırmaların kalitesi ve etkinliği de artırılmaktadır. Böylece hem arkeolojik alanlarda buluntular üzerinde çalışan araştırmacıların hem de dışarıdan bir araştırmacının bilimsel çalışmaları için tüm buluntulara kolay bir şekilde erişebileceği ve istediği bilgileri uygulama üzerinden kolay şekilde analiz ederek elde edebileceği bir platformun oluşturulmuş olması tezin sahaya olan katkısını da gösterir niteliktedir.

Genel olarak bu tez kapsamında yapılan çalışmalar değerlendirildiğinde; geliştirilen dijital envanter yönetim sistemi, arkeolojik buluntuların etkin bir şekilde kaydedilmesini, yönetilmesini ve analiz edilmesini sağlamaktadır. Kullanılan teknolojiler ve seçilen modeller, sistemin performansını ve kullanıcı deneyimini üst düzeye çıkarmaktadır. EfficientNetB0 modelinin seçimi, görüntü işleme ve derin öğrenme uygulamalarında yüksek performans ve verimlilik sunarak arkeolojik verilerin daha doğru ve hızlı bir şekilde işlenmesini sağlamaktadır.

Gelecekteki Çalışmalar için Öneriler

Model ve Algoritma Geliştirmeleri: Yapay zekâ ve makine öğrenmesi algoritmalarının kullanımını artırarak buluntuların otomatik sınıflandırılması ve analizi süreçleri iyileştirilmesi sistemin doğruluğunu ve hızını artırabilir.

Kullanıcı Geri Bildirimleri: Sistem, kullanıcı geri bildirimlerine dayanarak sürekli olarak güncellenebilir ve iyileştirilebilir. Kullanıcı deneyimini artırmak için arayüz ve işlevsellik geliştirmeleri yapılabilir.

Veri Entegrasyonu ve Genişletme: Sistem, farklı veri kaynakları ile entegre edilerek daha kapsamlı bir veri yönetimi sağlanabilir. Söz konusu entegrasyon arkeolojik araştırmaların kapsamını genişletebilir ve daha zengin bir veri analizi sunabilir.

Güvenlik ve Veri Koruma: Kullanıcı güvenliğini artırmak ve sistemin güvenilirliğini pekiştirmek için veri güvenliği ve gizliliği konusunda ek önlemler alınabilir.

Gelişmiş Görsel Veri İşleme: Buluntuların sınıflandırılması ve özelliklerinin belirlenmesinde daha iyi sonuçlar elde edilmesi için farklı makine öğrenmesi ve görüntü işleme teknikleri kullanılarak buluntuların görsel analizlerinin geliştirilmesi sağlanabilir.

6. KAYNAKLAR

- Abutaleb, H., Tamimi, A. ve Alrawashdeh, T. (2021). Empirical study of most popular PHP framework. *2021 International Conference on Information Technology (ICIT)*, 608-611. <https://doi.org/10.1109/ICIT52682.2021.9491679>
- Aguiar, E., Calumby, R., Oliveira, L., Santos, W. ve Duarte, A. (2021). Pathospotter-search: A content-based image retrieval tool for nephropathology. In *Anais Estendidos da XXXIV Conference on Graphics, Patterns and Images*, (pp. 146-149). Porto Alegre: SBC.
- Akbulut, M. ve Çare, B. (2011). Beytepe Kampüsü haritası mashup uygulaması. *Bilgi Dünyası*, 12(2), 334-346. <https://doi.org/10.15612/BD.2011.201>
- Alamsyah, N., Murpratiwi, S. I., Albar, M. A., Jatmiko, A. H., Huwae, R. B. ve Trieanto, A. (2024). Design and build a website-based accounting information system with extreme programming methods. *Journal of Computer Networks, Architecture and High Performance Computing*, 6(1), 270-283. <https://doi.org/10.47709/cnahpc.v6i1.3305>
- Alhakbani, N., Alghamdi, M. ve Al-Nafjan, A. (2023). Design and development of an imitation detection system for human action recognition using deep learning. *Sensors*, 23(24), 9889. <https://doi.org/10.3390/s23249889>
- Amin, H., Darwish, A., Hassanien, A. E. ve Soliman, M. (2022). End-to-end deep learning model for corn leaf disease classification. *IEEE Access*, 10, 31103-31115. <https://doi.org/10.1109/ACCESS.2022.3159678>
- Ananda, D., Hidayati, H. ve Cahyana, C. (2023). The development of human resources data management application at PT. Anugrah Laut Indonesia. *IJAIT (International Journal of Applied Information Technology)*, 1, 21-31. <https://doi.org/10.25124/ijait.v7i01.5982>
- Bagwan, K. I. ve Ghule, S. (2019). A modern review on Laravel- PHP Framework. *IRE Journals*, 2(12), 1-3.
- Baydogan, M. P., Baybars, S. C. ve Tuncer, S. A. (2023). Age-Net: An advanced hybrid deep learning model for age estimation using orthopantomograph images. *Traitement Du Signal*, 40(4), 1553-1563. <https://doi.org/10.18280/ts.400423>
- Bütüner, R., Şimşek, M., Aksöz, G. ve Şan, S. (2023). Yapay sinir ağları ile mısır yapraklarında hastalık tespiti. *Journal of Information Systems and Management Research*, 5(2), 21-31. <https://doi.org/10.59940/jismar.1384930>
- Cazenave, F., Quint, V. ve Roisin, C. (2011). Timesheets.js: When SMIL meets HTML5 and CSS3. *Proceedings of the 11th ACM Symposium on Document Engineering*, 43-52.
- Chyrvon, A., Lisovskyi, K. ve Kyryndas, N. (2023). The main methods of load balancing on the NGINX web server. *Collection of Scientific Papers «ΑΙΟΓΟΣ»*, (May 26, 2023; Boston, USA), 146-151. <https://doi.org/10.36074/logos-26.05.2023.040>
- De La Torre, I., Arroyo, A., Proffitt, T., Martín Ramos, C. ve Theodoropoulou, A. (2015). Archaeological fieldwork techniques in Stone Age sites. Some case studies. *Treballs d'Arqueologia*, 20, 21-40. <https://doi.org/10.5565/REV/TDA.51>

- Doğan, F. ve Türkoğlu, İ. (2019). Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. *DÜMF Mühendislik Dergisi*, 10(2), 409-445. <https://doi.org/10.24012/DUMF.411130>
- Efkleidou, K., Kaimaris, D., Roustanis, T., Patias, P., Andreou, S., Klimantakis, K., Karolos, I.-A., Pappa, M. ve Kouidis, N. (2022). Smart Eye: An application for In Situ Accessibility to “Invisible” heritage sites. *Journal of Computer Applications in Archaeology*, 5(1), 286-298. <https://doi.org/10.5334/jcaa.100>
- Gan, L., Huang, Y. ve Cheng, Y. (2022). Research on intelligent learning platform system based on Spring Boot. *2022 International Conference on Computer Science, Information Engineering and Digital Economy (CSIEDE 2022)*, Guangzhou, China.
- Gardner, A., Kanno, J., Duncan, C. A. ve Selmic, R. (2014). Measuring distance between unordered sets of different sizes. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 137-143. <https://doi.org/10.1109/CVPR.2014.25>
- Gidding, A., Matsui, Y., Levy, T. E., DeFanti, T. ve Kuester, F. (2013). ArchaeoSTOR: A data curation system for research on the archeological frontier. *Future Generation Computer Systems*, 29(8), 2117-2127. <https://doi.org/10.1016/j.future.2013.04.007>
- Greiff, M. ve Johansson, A. (2019). *Symfony vs express: A server-side framework comparison* (Bachelor's thesis, Blekinge Institute of Technology, Sweden). Kenneth Lewenhagen. Retrieved from <https://www.diva-portal.org/smash/get/diva2:1327290/FULLTEXT01.pdf>
- Gupta, S., Khanna, P., Kumar, S. ve Pragya. (2024). E-commerce website performance evaluation: Technology, strategy and metrics. *Asian Journal of Research in Computer Science*, 17(6), 114-125. <https://doi.org/10.9734/ajrcos/2024/v17i6461>
- Hanmandlu, M., Gupta, D. ve Vasikarla, S. (2013). Face recognition using Elastic bunch graph matching. *2013 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 1-7. <https://doi.org/10.1109/AIPR.2013.6749338>
- Henninger, M. (2018a). From mud to the museum: Metadata challenges in archaeology. *Journal of Information Science*, 44(5), 658-670. <https://doi.org/10.1177/0165551517741790>
- Henninger, M. (2018b). From mud to the museum: Metadata challenges in archaeology. *Journal of Information Science*, 44(5), 658-670. <https://doi.org/10.1177/0165551517741790>
- Henshaw, R. ve Valauskas, E. J. (2001). Metadata as a Catalyst: Experiments with metadata and search engines in the internet journal, first monday. *Libri*, 51(2), 86-101. <https://doi.org/10.1515/LIBR.2001.86>
- Ibañez, V., Bohlen, L., Manuella, F., Mansuy, I., Helmchen, F. ve Wahl, A.-S. (2023). EXPLORE: A novel deep learning-based analysis method for exploration behaviour in object recognition tests. *Scientific Reports*, 13(1), 4249. <https://doi.org/10.1038/s41598-023-31094-w>
- Jiang, L., Liu, H., Jiang, H., Zhang, L. ve Mei, H. (2022). Heuristic and neural network based prediction of project-specific API member access. *IEEE Transactions on Software Engineering*, 48(4), 1249-1267. <https://doi.org/10.1109/TSE.2020.3017794>

- Kabasakal, İ. ve Soyuer, H. (2021). A Jaccard similarity-based model to match stakeholders for collaboration in an industry-driven portal. *Proceedings*, 74(1), 15. <https://doi.org/10.3390/proceedings2021074015>
- Keskin, Y. ve Tanaç Zeren, M. (2022). Arkeolojik alanlarda mimari düzenlemelerin ve sunum tekniklerinin Türkiye'deki örnekler üzerinden karşılaştırmalı değerlendirilmesi. *Artium*, 10(1), 21-35. <https://doi.org/10.51664/artium.948113>
- Lange, D., Judson-Torres, R., Zangle, T. ve Lex, A. (2024). *Aardvark*: composite visualizations of trees, time-series, and images. *IEEE*. <https://doi.org/10.31219/osf.io/cdbm6>
- Mardianto, I., Sugianto, D. ve Ashari, K. A. (2021). Elastic stack ability test monitoring slowloris attack on digital ocean server. *Ultimatics: Jurnal Teknik Informatika*, 13(2), 120-126. <https://doi.org/10.31937/ti.v13i2.2209>
- Masih, E. A. (2022). Feasibility of using QR code for registration & evaluation of training and its ability to increase response rate – The learners' perception. *Nurse Education Today*, 111, 105305. <https://doi.org/10.1016/j.nedt.2022.105305>
- Moore, D. (2024). *Components of JWTs explained*. Retrieved from <https://fusionauth.io/articles/tokens/jwt-components-explained>
- Niarman, A., Iswandi ve Candri, A. K. (2023). Comparative analysis of PHP frameworks for development of academic information system using load and stress testing. *International Journal Software Engineering and Computer Science (IJSECS)*, 3(3), 424-436. <https://doi.org/10.35870/ijsecs.v3i3.1850>
- Niewolski, W., Nowak, T. W., Sepczuk, M. ve Kotulski, Z. (2021). Token-based authentication framework for 5G MEC mobile networks. *Electronics*, 10(14), 1724. <https://doi.org/10.3390/electronics10141724>
- Norhaidah, A. (2019). PHP frameworks usability in web application development. *International Journal of Recent Technology and Engineering*, 8(3S), 109-116. <https://doi.org/10.35940/ijrte.C1020.1083S19>
- Obilikwu, P. ve Terwase, V. (2021). An implementation of Replication Oriented Architecture (ROA) for web service scalability. *Nigerian Annals of Pure and Applied Sciences*, 4(1), 168-180. <https://doi.org/10.46912/napas.176>
- Orhan, O., Yakar, M., ve Kirtiloglu, O. S. (2017). A web based service application for visual sinkhole inventory information system; case study of Konya closed basin. *Selcuk University Journal of Engineering, Science and Technology*, 5(1), 72-82. <https://doi.org/10.15317/Scitech.2017.71>
- Özcan, A. (2024). Konya Arkeoloji Müzesi'nden bir grup metal eser. *OANNES - Uluslararası Eskiçağ Tarihi Araştırmaları Dergisi*, 6(1), 105-135. <https://doi.org/10.33469/oannes.1399701>
- Porebski, B., Przystalski, K. ve Nowak, L. (2011). *Building PHP applications with Symfony, CakePHP, and Zend Framework*. John Wiley and Sons.
- Putra, O. V., Suryadarma, Y. ve Maulana, M. A. (2021). Design of Management Information Systems Memorandum of Understanding (MoU) and Memorandum of Agreement (MoA) Universitas Darussalam Gontor. *Applied Technology and Computing Science Journal*, 4(1), 16-25. <https://doi.org/10.33086/atcsj.v4i1.2152>

- Roosevelt, C. H., Cobb, P., Moss, E., Olson, B. R. ve Ünlüsoy, S. (2015). Excavation is destruction digitization: Advances in archaeological practice. *Journal of Field Archaeology*, 40(3), 325-346. <https://doi.org/10.1179/2042458215Y.0000000004>
- Sachdeva, V., Freimuth, D. M. ve Mueller, C. (2009). Evaluating the jaccard-tanimoto index on multi-core architectures. In *Computational Science–ICCS 2009: 9th International Conference Baton Rouge, LA, USA, May 25-27, 2009 Proceedings, Part I 9* (pp. 944-953). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-01970-8_95
- Sendiang, M., Kasenda, S., Polii, A. ve Putung, Y. (2018). Optimizing Laravel authentication process. In *2018 International Conference on Applied Science and Technology (ICAST)* (pp. 247-251). IEEE.
- Siame, A. ve Kunda, D. (2017). Evolution of PHP applications: A systematic literature review. *International Journal of Recent Contributions from Engineering, Science & IT (IJES)*, 5(1), 28-39. <https://doi.org/10.3991/ijes.v5i1.6437>
- Singh, R. H., Maurya, S., Tripathi, T., Narula, T. ve Srivastav, G. (2020). Movie recommendation system using cosine similarity and KNN. *International Journal of Engineering and Advanced Technology*, 9(5), 556-559. <https://doi.org/10.35940/ijeat.E9666.069520>
- Stauffer, M. (2023). *Laravel: Up & running*. O'Reilly.
- Steven, S., Rifaldi, W. ve Nugraha, U. (2023). Strategi Pengamanan Front-end dalam Pengembangan Website. *JUSTINFO | Jurnal Sistem Informasi dan Teknologi Informasi*, 1(1), 42-53. <https://doi.org/10.33197/justinfo.vol1.iss1.2023.1200>
- Sukestiyarno, Y. L., Sapolo, H. A. ve Sofyan, H. (2023). *Application of recommendation system on E-learning platform using content-based filtering with jaccard similarity and cosine similarity algorithms. Preprints.* <https://doi.org/10.20944/preprints202306.1672.v1>
- Sukma, F., Hadi, R., Chairani, F., Sariani ve Wardhani, A. (2023). Information system and population data search using the Boyer-Moore Algorithm in Nagari Kotobaru Simalanggang. *International Journal of Advanced Science Computing and Engineering*, 5(3), 305-316. <https://doi.org/10.62527/ijasce.5.3.185>
- Šušter, I. ve Ranisavljević, T. (2023). Optimization of MySQL database. *Journal Of Process Management and New Technologies*, 11(1-2), 141-151. <https://doi.org/10.5937/jpmnt11-44471>
- Taner, A., Öztekin, Y. B. ve Duran, H. (2021). Performance analysis of deep learning CNN models for variety classification in Hazelnut. *Sustainability*, 13(12), 6527. <https://doi.org/10.3390/su13126527>
- Tezer Altay, S., Ateş, G. ve Mania, U. (2022). Archaeology meets local needs? Experiences of the pergamon excavation in intensifying local awareness of cultural heritage and its protection. *Türkiye Bilimler Akademisi Kültür Envanteri Dergisi*, 25, 33-52. <https://doi.org/10.22520/tubaked2022.25.007>
- Tiwari, S. (2016). An introduction to QR code technology. *2016 International Conference on Information Technology (ICIT)*, 39-44. <https://doi.org/10.1109/ICIT.2016.021>
- Yang, J. A. ve Aklani, S. A. (2023). Performance analysis between interpreted language-based (laravel) and compiled language-based (GIN) web frameworks. *Computer*

Based Information System Journal, 11(1), 12-16.
<https://doi.org/10.33884/cbis.v11i1.6583>

- Yousif, A. N. ve Ibrahim, H. M. (2023). Online destinations map using Google Maps API based on the private database. *International Journal of Information Technology and Computer Engineering*, 35, 35-39. <https://doi.org/10.55529/ijitc.35.35.39>
- Zou, Y., Li, J. ve Chen, X. (2017). Seam tracking investigation via striped line laser sensor. *Industrial Robot: An International Journal*, 44(5), 609-617. <https://doi.org/10.1108/IR-11-2016-0294>
- Zou, Y., Tang, P. ve Lv, T. (2022). Design and implementation of ship shore power information management system based on Spring Boot and Vue. *Frontiers in Science and Engineering*, 2(12), 17-22. <https://doi.org/10.54691/fse.v2i12.3284>

