

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**OSMANLICA ESERLER ÜZERİNDE SIFT
ALGORİTMASI İLE MOBİL UYGULAMA GELİŞTİRME**
(Yüksek Lisans Tezi)

Tezi Hazırlayan:
Abdullah GÜRSOY

İstanbul, 2019

T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ BİLİM DALI

**OSMANLICA ESERLER ÜZERİNDE SIFT
ALGORİTMASI İLE MOBİL UYGULAMA GELİŞTİRME**
(Yüksek Lisans Tezi)

Tezi Hazırlayan:

Abdullah GÜRSOY

Öğrenci No:

150820054

Danışman:

Dr. Öğr. Üyesi Ediz ŞAYKOL

İstanbul, 2019

YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Osmanlıca Eserler Üzerinde SIFT Algoritması İle Mobil Uygulama Geliştirme” başlıklı bu çalışmanın, bilimsel ahlak ve geleneklere uygun şekilde yazdığımı, yararlandığım eserlerin tamamının kaynaklarda gösterildiğini ve çalışmanın içinde kullandıkları her yerde bunlara atıf yapıldığını belirtir ve bunu onurumla doğrularım. 17/06/2019

Abdullah GÜRSOY



T.C.
BEYKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

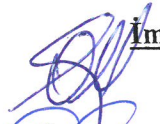
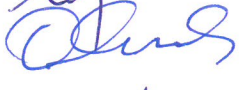

YÜKSEK LİSANS TEZ SAVUNMA SINAVI SONUÇ TUTANAĞI

Beykent Üniversitesi
Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Aşağıda tez adı belirtilen yüksek lisans öğrencisi.....no'luin tarihinde yapılan tez savunma sınavı¹ sonucunda..... dakika süreyle sunduğu ve savunduğu tezi hakkında² oybirliğiyle, **KABUL** kararı verilmiştir.

Bilgilerinize saygılarımızla arz ederiz.

Anabilim Dalı : Fen Bilimleri
Programı : Bilişimci Mühendisliği
Tez Başlığı³ : Osmanlıca Eseler Üzerinde SIFT Algoritması ile Mobil Uygulama Geliştirme

<u>Tez Sınav Jürisi</u>	<u>Öğretim Üyesi</u>	<u>İmza</u>
Danışman	: Dr. Öğr. Üyesi Ediz ŞAYKOL	
Üye	: Dr. Öğr. Üyesi Ömer GENÇ	
Üye	: Dr. Öğr. Üyesi ATILG YILMAZ	

¹ Jüri üyeleri, söz konusu tezin kendilerine teslim edildiği tarihten itibaren en geç bir ay içinde toplanarak öğrenciyi tez sınavına alır. Tez savunma sınav süresi en az 45, en çok 90 dakikadır. Jüri üyeleri, sınav öncesi yapılacak toplantıda, kendi aralarından danışman dışında bir üyeyi başkan seçer. Tez sınavı, tez çalışmasının sunulması ve bunu izleyen soru-cevap bölümünden oluşur. Tez sınavı, öğretim elemanları, lisansüstü öğrenciler ve alanın uzmanlarından oluşan dinleyicilerin katılımına açık ortamlarda gerçekleştirilir. Belirlenen günde yapılamayan jüri toplantısı, katılanların hazırladığı bir tutanakla enstitü yönetimine bildirilir. Bu durumda, jüri en geç on beş gün içinde toplanarak adayı tez savunma sınavına alır. (05 Ağustos 2017 tarihli 30145 sayılı Resmi Gazetede Yayınlanan Değişiklik-Madde 29-3)

² Tez sınavının tamamlanmasından sonra jüri, tez hakkında salt çoğunlukla “kabul”, “düzeltme” veya “ret” kararı verir. Jüri başkanı, jüri üyelerince imzalanmış karar tutanağını, tez sınavını izleyen üç gün içinde ilgili enstitü yönetimine teslim eder. Tezi hakkında düzeltme kararı verilen öğrenci en geç üç ay içinde gerekli düzeltmeleri yaparak ve birinci fıkradaki usule göre tezini aynı jüri önünde yeniden savunur. Süresi içerisinde “düzeltme” savunmasına girmeyen öğrencinin enstitü ile ilişkisi kesilir. (Beykent Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği-Madde 29-4)

³ İleride doğabilecek aksaklıkların engellenmesi için tezin başlığının yazılması gerekmektedir.

Adı ve Soyadı : Abdullah GÜRSOY
Danışmanı : Dr. Öğr. Üyesi Ediz ŞAYKOL
Türü ve Tarihi : Yüksek Lisans, 2019
Alanı : Bilgisayar Mühendisliği
Anahtar Kelimeler : SIFT Algoritması, Resim Eşleme, Osmanlıca, Kitabe, Hat, Şahide

ÖZ

OSMANLICA ESERLER ÜZERİNDE SIFT ALGORİTMASI İLE MOBİL UYGULAMA GELİŞTİRME

Günümüze ulaşmış bir çok Osmanlıca eser mevcuttur. Bu eserlerin bilgilerine erişim ve üzerindeki Osmanlıca yazılarının okunması oldukça zordur. Bu tez çalışması içerisinde öncelikle; okunamayan eserler tanımlanmış daha sonra bu eserlerin bir mobil uygulama sayesinde bilgilerine erişilmesine ve günümüz Türkçe'si ile okunmasına SIFT algoritmasına GPS tabanlı filtre uygulanarak olanak sağlanmıştır.

Ayrıca bu eserlerin ileride akademik çalışmalarda kullanılması için bir envanter platformu oluşturulmuştur. Tez kapsamında belirlenmiş pilot bölgelerdeki eserler daha önce İçerik Yönetim Sistemine girilmiş; SIFT algoritması ile gerekli bilgiler çıkartılarak daha sonraki aramalar için kaydedilmiştir.

Daha sonra geliştirilen mobil uygulama üzerinden resim tanıma sistemine yapılan arama isteklerine eşleşen resimler; içerik yönetim sistemindeki bilgiler ile birleştirilerek kullanıcıya iletilmiştir.

Name and Surname : Abdullah GÜRSOY
Supervisor : Dr. Lecturer Ediz ŞAYKOL
Türü ve Tarihi : Master, 2019
Alanı : Computer Engineering
Anahtar Kelimeler : SIFT Algorithm, Image Matching, Ottoman Artifacts,
Inscriptions,

ABSTRACT

A MOBILE APPLICATION ON OTTOMANS ARTIFACTS USING SIFT ALGORITHM

Many Ottoman artifacts have reached to today. Accessing information of those artifacts and reading Ottoman writings on it have difficulties. In this thesis, first; the unreadable artifacts were defined and a mobile application was created to enable access to their information and translations to current Turkish language by applying GPS based filtering to SIFT algorithm.

In addition, an inventory platform has been created for future use of these artifacts in academic studies. The artifacts in the pilot regions identified within the scope of this thesis were previously entered into the Content Management System. All the artifacts were sent to Image Recognition System to extract important points with SIFT algorithm. Then those important points were saved to database for subsequent calls.

Finally, using the mobile application to send search requests to Image Recognition service, the end user gains access to the images matched with the search request and the relative information gathered from the Content Management System.

İÇİNDEKİLER

ÖZ.....	i
ABSTRACT.....	ii
ŞEKİLLER LİSTESİ.....	v
1. GİRİŞ	1
2. GENEL MİMARİ	2
3. SIFT ALGORİTMASI	4
3.1 Ölçek(Yakınlık-Uzaklık) Uzayı.....	4
3.2 Ölçekler Arasındaki Farkların Elde Edilmesi.....	5
3.3 Maksimum ve minimum değerlerin çıkartılması.....	6
3.4 Maksimum ve Minimum Değerlerin Resim Üzerindeki Koordinatların Belirlenmesi	7
3.5 Düşük Kontrastlı Noktaların Elenmesi.....	8
3.6 Kenarların Silinmesi	8
3.7 Önemli Noktaların Gradyan Yön Ağırlıklarının Belirlenmesi.....	9
3.8 Önemli Noktaların İmzasının Çıkartılması.....	11
3.9 Parlaklıktan bağımsızlığın eklenmesi.....	12
3.10 SIFT Algoritmasının tercih sebebi	12
4. MOBİL UYGULAMA İÇİN SIFT ALGORİTMASI.....	13
4.1 Konum Tabanlı Optimizasyon.....	13
4.2 Apache Lucene Arama Servisi.....	13
4.3 Resimlerin Depolanması.....	14
4.4 SIFT Algoritmasının Rest Servise Çevrilmesi.....	15
4.4.1 Resim Ekleme Servisi.....	15
4.4.2- Resim Arama Servisi	15
4.4.3 - Resim Silme Servisi.....	17
4.4.4 - Resim Adedi Servisi.....	17
4.4.5 - Resim Kontrol Servisi.....	17
4.4.6 - Resim Tanıma Durum Servisi	18
5. İÇERİK YÖNETİM SİSTEMİ.....	19
5.1 Rol Yönetimi	19
5.2 Eserlerin Tanımlanması	20
5.2.1 Kitabe	20
5.2.2 Hat Eserleri.....	21
5.2.3 Şahide	22

6. İÇERİK YÖNETİM SİSTEMİ İLE RESİM TANIMA SİSTEMİ ENTEGRASYONU	24
7. İÇERİK YÖNETİM SİSTEMİ İLE MOBİL UYGULAMA ENTEGRASYONU	26
8. MOBİL UYGULAMA VE TASARIM ÖRNEKLERİ	27
9. TEST	28
10. SONUÇ	30
KAYNAKÇA	32
EKLER	35
Ek-1: Mobil uygulama kayıt ve giriş ekranı	35
Ek-2: Mobil uygulama eser arama ekranı	35
Ek-3 Mobil uygulama eşlenmiş eser ekranları	36
Ek-4 Mobil uygulama bina ve eser detayları ekranları	36
Ek-5 Mobil uygulama rota ekranları	37
Ek-6 İçerik Yönetim Sistemi Ana Ekran	37
Ek-7 İçerik Yönetim Sistemi Bina Ekle Ekranı	38
Ek-8 İçerik Yönetim Sistemi Kitabe Ekle Ekranı	39
Ek-9 İçerik Yönetim Sistemi Hat Ekle Ekranı	40
Ek-10 İçerik Yönetim Sistemi Şahide Ekle Ekranı	41
Ek-11 İçerik Yönetim Sistemi Kişi Ekle Ekranı	42
Ek-12 İçerik Yönetim Sistemi Çoktan Seçmeli Soru Ekle Ekranı	42
Ek-13 İçerik Yönetim Sistemi Doğru Yanlış Soru Ekle Ekranı	43

ŞEKİLLER LİSTESİ

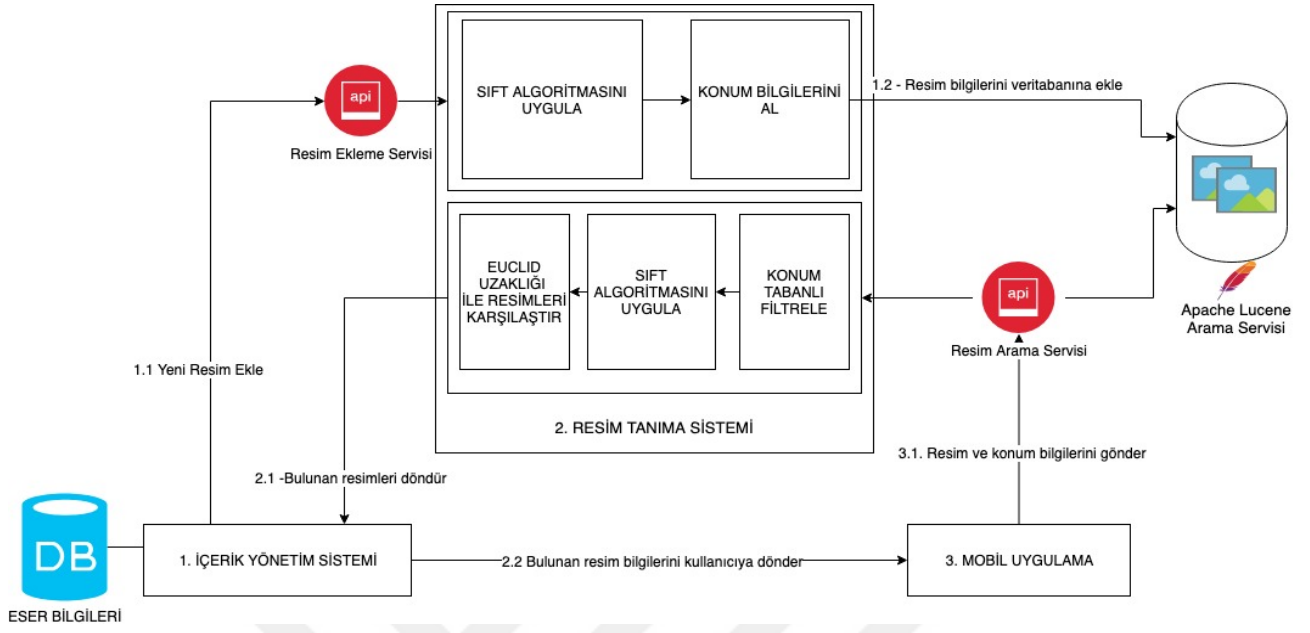
Şekil 1. Sistemin mimarisine üst bakış	3
Şekil 3. Gauss farkları bulunmuş resim örnekleri.....	6
Şekil 4. Minimum ve maksimum olabilecek adayların bulunması.....	6
Şekil 5. Gerçek koordinatların tesbit edilmesi.....	7
Şekil 6. Düşük kontrastlı noktalar elendikten sonraki önemli noktalar	8
Şekil 7. Düşük kontrastlı noktalar elendikten sonraki önemli noktalar	9
Şekil 8. Gradyan Yön Ağırlıklarının Histogram Gösterimi.....	10
Şekil 9. Yön ağırlıklarının resim üzerinde gösterilmesi	10
Şekil 10. Önemli noktaların imzalarının çıkartılması	11
Şekil 11. Amazon S3 üzerinden resim datası akışı	14
Şekil 12. Sünbül Efendi Camii'nde bir kitabe örneği	21
Şekil 13. Mahmut Celaleddin Efendi'ye ait bir hat levhası	22
Şekil 14. Fatih Camii içerisindeki bir şahide örneği.....	23
Şekil 15. İçerik ekleme, silme ve güncelleme senaryosu temsili gösterim.....	24
Şekil 16. Resim Tanıma Sisteminin gerçek veriler ile test edilmesi. Sağdaki aranan resim, solda doğru bulunan sonuç	28
Şekil 17. Resim Tanıma Sisteminin gerçek veriler ile test edilmesi 2. örnek Sağdaki aranan resim, solda doğru bulunan sonuç.....	28
Şekil 18. Resim Tanıma Sisteminin gerçek veriler ile test edilmesi 3. örnek. Sağdaki aranan resim, solda doğru bulunan sonuç.....	29
Şekil 19. Resim Tanıma Sisteminin gerçek veriler ile test edilmesi 4. örnek. Sağdaki aranan resim, solda doğru bulunan sonuç.....	29

1. GİRİŞ

Başta İstanbul olmak üzere Türkiye ve Osmanlı Devleti idaresi altında bulunmuş bir çok ülkede Osmanlıca eserler mevcuttur. Bu eserler akademisyenler tarafından çalışılmalar yapılmış fakat farklı farklı akademik çalışmalar içerisinde yer aldığı için bu bilgilere erişim oldukça zordur. Ayrıca üzerindeki Osmanlıca metinler, eserin ait olduğu dönemin dil özelliğini içerdiğinden bu metinleri okumak ayrı bir uzmanlık gerektirmektedir. Bu tez çalışması içerisinde öncelikle; okunamayan eserler tanımlanmış daha sonra bu eserlerin bir mobil uygulama sayesinde bilgilerine erişilmesine ve günümüz Türkçe'si ile okunmasına SIFT algoritmasına konum tabanlı filtre uygulanarak olanak sağlanmıştır. Ayrıca bu eserlerin tanımlanması ve bu sahada akademik alanda çalışma yapanlar tarafından kullanmak üzere bir envanter platformu oluşturulmuştur. Başlangıç olarak pilot bölgelerdeki eserler tanımlanmış. Ardından envanter platformuna bilgi girişleri sağlanmıştır. Bilgi girişleri esnasında her bir eser için SIFT algoritması ile gerekli bilgiler çıkartılarak daha sonraki aramalar için kaydedilmiştir. SIFT algoritmasının bir mobil uygulama servisi olabilmesi ve performansının artırılmasına yönelik yaptığımız bazı optimizasyonlar ileride anlatılacaktır. Daha sonra geliştirdiğimiz mobil uygulama üzerinden resim tanıma sistemine yapılan arama isteklerine eşleşen resimler; içerik yönetim sistemindeki bilgiler ile birleştirilerek kullanıcıya sunulmuştur.

2. GENEL MİMARİ

Tez kapsamında üç ayrı modül oluşturulmuştur. Bunlardan biri eser bilgilerinin saklanacağı İçerik Yönetim Sistemi'dir. Diğer modül resim arama ve resim bilgilerinin tutulacağı Resim Tanıma Sistemi'dir. Üçüncü modül ise Resim Tanıma Sistemi'nin bir istemcisi olarak çalışacak Mobil Uygulama'dır. Şekil 1.'de belirtildiği gibi; İçerik Yönetim Sistemi'ne yeni bir eser eklendiğinde öncelikle bilgilere veritabanında saklanılacaktır. Ardından Resim Tanıma Sistemi'nin 'resim ekle' servisine, eklenen eserlerin resimleri ve enlem boylam bilgileri gönderilecektir. Resim Tanıma Sistemi herbir resim için öncelikle SIFT algoritmasını uygulayacaktır. Daha sonra SIFT algoritmasının sonucundan üretilen tanımlıyıcılar(descriptor) Apache Lucene Arama Servisine kaydedilecektir. Bu işlemler gelen herbir resim için uygulandıktan sonra arama servisine aynı dosya isimleri ile kaydedilecektir. Bir eserin birden fazla resiminin bulunması o eserin arandığında bulunma oranını arttırmaktadır. Bir eserin birden fazla resminin olması dosya isimlerindeki anahtarın aynı şekilde olması ile sağlanmıştır. Mobil Uygulama tarafından bir resim aranacağı zaman öncelikle Resim Tanıma Sistemi'ne aranan resimin verileri ve enlem, boylam bilgileri gönderilecektir. Resim Tanıma Sistemi öncelikle; enlem ve boylam bilgilerine yakın olan resimleri filtreleyecektir. Daha sonra aranacak resim üzerinde SIFT algoritmasını uygulayacaktır. SIFT algoritmasının sonucundan elde edilen veriler; Apache Lucene Arama Servisi üzerinden Euclid Uzaklık algoritması ile karşılaştırılarak eşleşen resimlerin listesini getirecektir. Daha sonra her bir eşleşen resimin bilgilerini İçerik Yönetim Sistemi'nden talep ederek mobil uygulamaya geri döndürülecektir. Böylece mobil uygulama eşleşen resimlere ve bilgilere ulaşmış olacaktır.



Şekil 1. Sistemin mimarisine üst bakış

3. SIFT ALGORITMASI

David Lowe ve ekibinin geliştirdiği SIFT görüntü eşleme algoritması[1] bir görüntünün eğikliği, yakınlığı, açı değişimi, karanlık veya aydınlık farkından bağımsız resime özel bir imzanın üretilmesini hedeflemektedir.

Bu bağımsızlığın sağlanması için seçilen algoritma tarafından belirlenen ‘önemli noktalar(interest points)’lerin etrafındaki gradyan yönelimleri ve büyüklükleri hesaplanmaktadır. Bu sayede görüntü üzerindeki açı değişse dahi gradyan yönelimindeki farkların değişmemesinden dolayı bağımsız sonuçlar elde edilebilmektedir. Açı, eğim ve aydınlık-karanlık değişimi parametrelerindeki değişimler, yakınlık ve uzaklık parametreleri ile doğrudan alakalı olduğundan öncelikle resimin farklı ölçeklerdeki simülasyonları oluşturulur ve ölçekler arasında ortak bir uzay oluşması sağlanır.

3.1 Ölçek(Yakınlık-Uzaklık) Uzayı

SIFT algoritmasının ölçek(yakınlık-uzaklık) değişimlerinden bağımsız çalışması için öncelikle bir ölçek uzayı oluşturulmaktadır. Bu uzayın elde edilmesi için ‘ölçek’ ve ‘oktav’ adında iki parametre oluşturulmaktadır. Bir resimin kaç defa yeniden boyutlandırılacağı belirlendiği parametre ‘oktav’ parametresidir. Her bir boyutlandırmada resimin ne kadar bulanıklaştırılacağı ise ‘ölçek’ parametresi ile belirlenir. SIFT algoritması bulanıklaştırma işlemi için Gauss Konvolsiyonu yönteminden faydalanmaktadır. Bu konvolsiyonun matematiksel eşitliği aşağıda belirtilmiştir Eşitlik 3.1:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (3.1)$$

Kaynak: Janez Krizaj, V.S., Nikola Pavešić (2010), *Adaptation of SIFT Features for Robust Face Recognition*. S. 3

Yukarıdaki eşitlikte L bulanık görüntüyü, G ise Gauss Konvolsiyon operatörünü ifade etmektedir. I ile ifade edilen ise orijinal resmin belirtilen koordinatlarındaki pixel datasıdır. Eşitlik içerisinde kullanılan parametrelerden x ve y resimin koordinatını, σ ise bulanıklaştırma katsayısını ifade etmektedir. Yukarıda belirtilen Gauss Konvolsiyon’unun açılımı ise aşağıdaki gibidir.

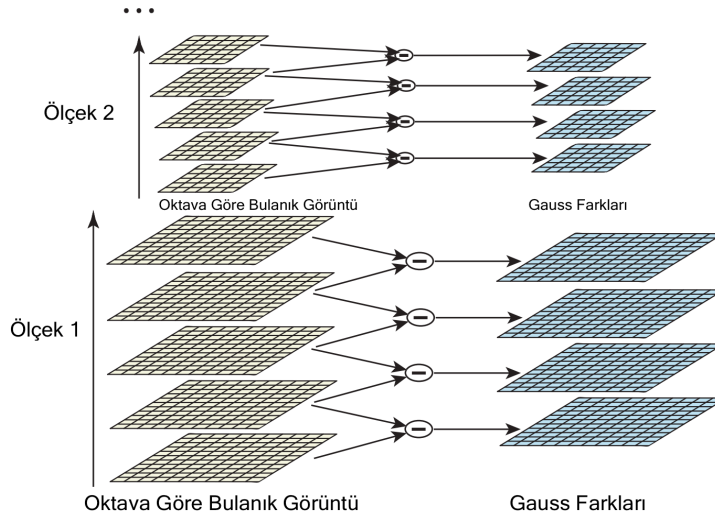
Eşitlik 3.2:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{-(x^2+y^2)/2\sigma^2} \quad (3.2)$$

Kaynak: Janez Križaj, V.S., Nikola Pavešić (2010), *Adaptation of SIFT Features for Robust Face Recognition*. S. 3

3.2 Ölçekler Arasındaki Farkların Elde Edilmesi

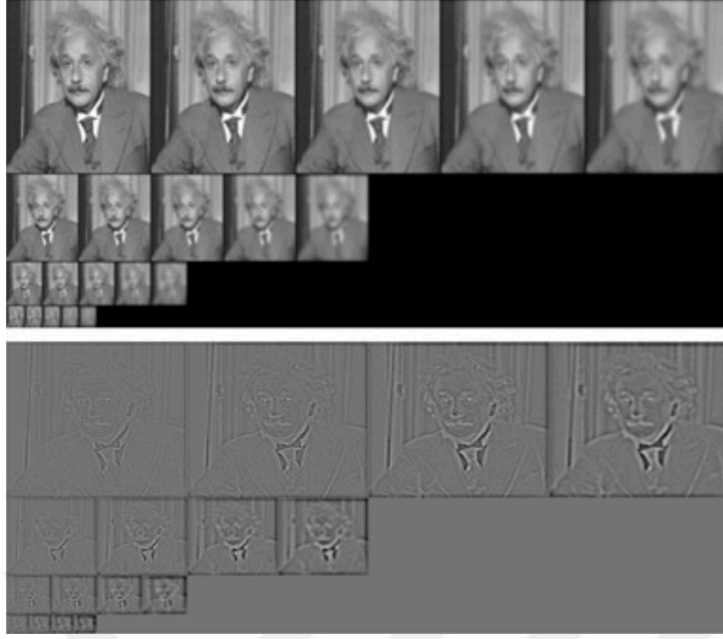
Gauss Konvolsiyonu ile elde edilmiş farklı ölçeklerdeki ve farklı bulanıklaştırma katsayılarındaki resimler öncelikle aşağıdaki şemadaki gibi listelenecektir. Daha sonra her bir oktavadaki resimlere sıralı çıkartma işlemi uygulanacaktır. Sıralı çıkartma işleminden sonra elde edilmiş resimler Gauss farklarını (Difference of Gauss)'u bulmak için kullanılacaktır.



Şekil 2. Gauss farkları temsili gösterim

Kaynak: Sinha, U. (2018). *SIFT: Theory and Practice*

Şekil 3.'de farklı ölçeklerde bulanıklaştırılmış resimlerin sıralı çıkartma işleminden sonraki görüntüleri gösterilmektedir.



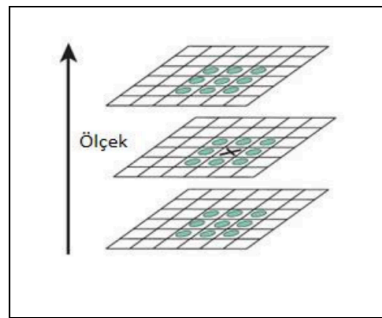
“

Şekil 3. Gauss farkları bulunmuş resim örnekleri

Kaynak: F. Estrada, A.J., D. Fleet (2004) *Local Features Tutorial*

3.3 Maksimum ve minimum değerlerin çıkartılması

Maximum değerlerin çıkartılması için Gauss farkları bulunan resimlerden 3'ürlü 2 set oluşturulur. Bu set; (1,2,3) ve (2,3,4)'üncü resimlerden oluşur. Daha sonra her bir setin orta ölçeği(2. ölçek ve 3. ölçek)'inde ki her bir pixel; alt ve üst ölçeklerinden 26 komşusu ile karşılaştırılır. Eğer seçilen nokta diğer 26 komşusu arasından en küçük veya en büyük değere sahip ise maksimum ve minimum olabilecek adaylar listesine eklenir. Bu işlem diğer ölçeklere de aynı şekilde uygulanır. Bakınız Şekil 4.



Şekil 4. Minimum ve maksimum olabilecek adayların bulunması

Kaynak: Sinha, U. (2018). *SIFT: Theory and Practice*

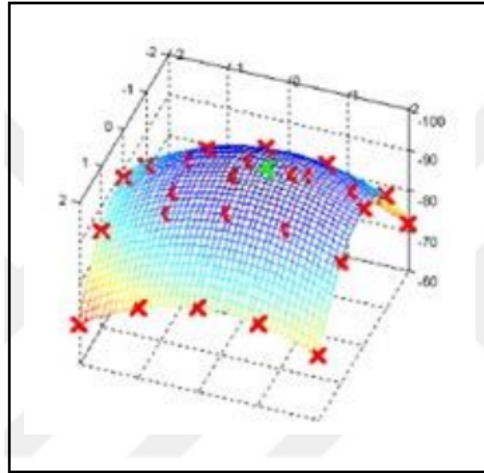
3.4 Maksimum ve Minimum Değerlerin Resim Üzerindeki Koordinatların Belirlenmesi

Her bir ölçekte bulunan önemli nokta(interest point) adayları gerçek resim üzerindeki koordinatı tam olarak ifade etmediği için; gerçek koordinatların bulunması için Sub-pixel Localization yöntemi SIFT algoritması içerisinde uygulanmaktadır. [1] Gerçek koordinatın bulunması için Gauss formülünün türevi alınarak 0'a eşitlenerek gerçek koordinatın tespit edilmesi sağlanmıştır. Bu işlem süresi çok zaman alacağından dolayı bu formülün Taylor Açılımı karşılığı ile bu işlem gerçekleştirilmektedir. Taylor açılımının matematiksel formülü aşağıda belirtildiği gibidir.

$$D(x) = D + \frac{\theta D^T}{\theta x} x + \frac{1}{2} x^T \frac{\theta^2 D}{\theta x^2} x$$

Kaynak: Fisher, R.B.(2016), *Subpixel Estimation, in Computer Vision*

Verilen maksimum ve minimum değerlerden gerçek koordinatların tesbit edilmesi aşağıdaki gibidir.

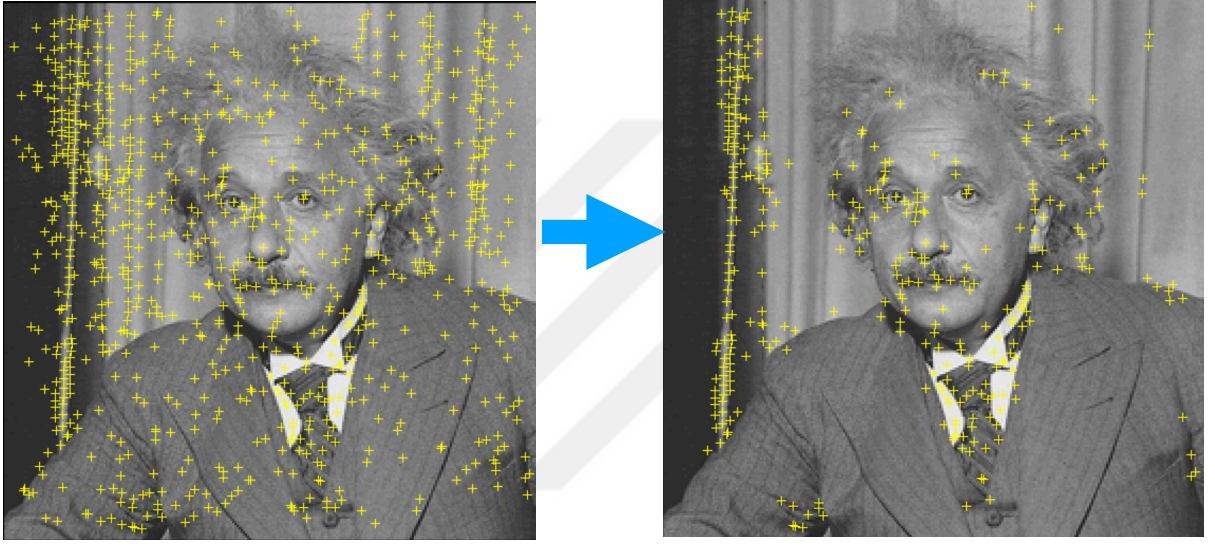


Şekil 5. Gerçek koordinatların tesbit edilmesi

Kaynak: Clemons, J. *SIFT: SCALE INVARIANT FEATURE TRANSFORM BY DAVID LOWE.*

3.5 Düşük Kontrastlı Noktaların Elenmesi

SIFT algoritmasının uyguladığı ilk eleme yöntemi olan bu işlemde düşük kontrastlı önemli noktalar elenmektedir.[1] Bir önceki adımda belirlenen önemli noktaların gerçek resim üzerindeki koordinatları belirlenmişti, bu adımda ise bu değerler çevresindeki değerler ile değerlendirilir ve düşük kontrast değerine sahip oldukları verilen eşik değerinden düşük olduğu tesbit edilirse; önemli noktalar listesinden silinmektedir. Bu filtrelemenin ardından elde edilen önemli noktaların gösterimi Şekil 6’da gösterilmektedir..

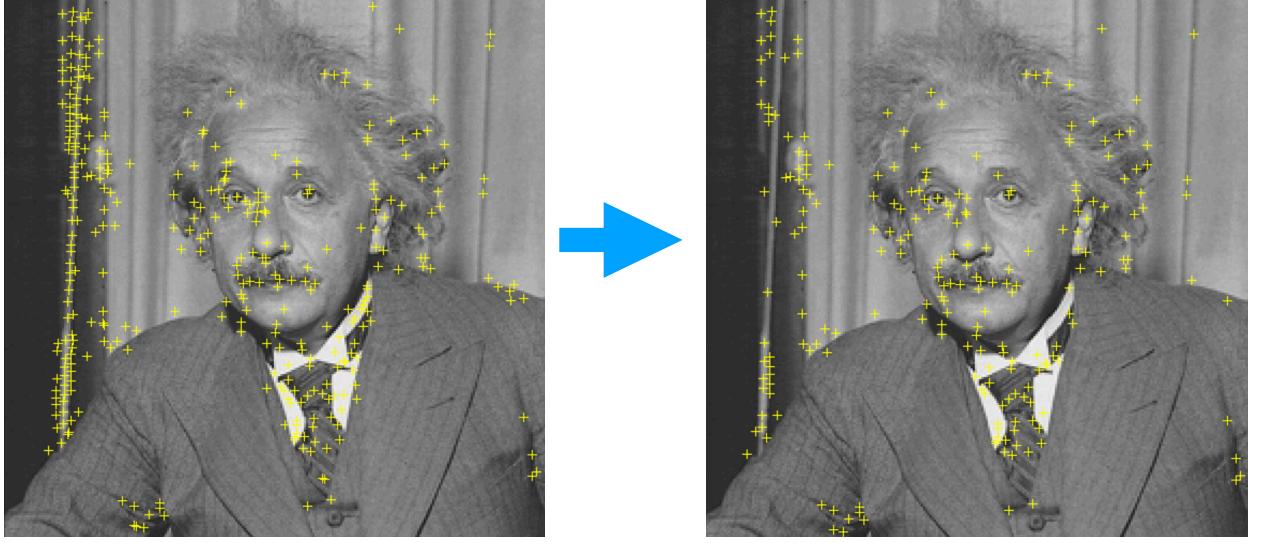


Şekil 6. Düşük kontrastlı noktalar elendikten sonraki önemli noktalar

Kaynak: F. Estrada, A.J., D. Fleet (2004) *Local Features Tutorial*

3.6 Kenarların Silinmesi

SIFT algoritması kenarların tesbit edilmesi için Harris köşe tesbiti(Harris Corner Detector) algoritmasından esinlenmiştir.[1] Bu algoritmaya göre belirtilen çerçevedeki alanın bir kenar mı, bir köşe mi yoksa düz bir zemin olduğu belirlenir. [7] Daha sonra belirlenen eşik değerine göre önemli noktalar arasında eleme yapılır. SIFT algoritmasının kenarları önemli noktalar listesinden silmesinden sonra elde edilen sonuç Şekil 7.’de gösterilmiştir.



Şekil 7. Düşük kontrastlı noktalar elendikten sonraki önemli noktalar

Kaynak: F. Estrada, A.J., D. Fleet (2004) *Local Features Tutorial*

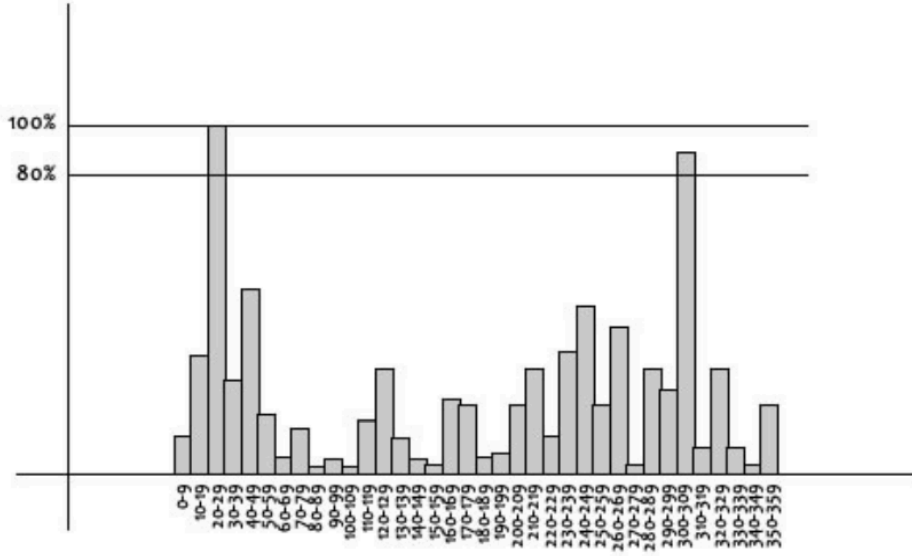
3.7 Önemli Noktaların Gradyan Yön Ağırlıklarının Belirlenmesi

Uygulanan filtrelerden sonra elde edilmiş önemli noktaların(interest point), açı ve yön değişkenlerinden bağımsız olması için SIFT algoritması Gradyan Yön Ağırlıkları(Gradient Magnitudes) metodunu kullanmaktadır.[1] Bu metodda her bir önemli nokta çevresindeki diğer pixel dataları ile birlikte hangi yöne ne kadar yoğunlukta hareket gösterdiği elde edilmektedir. Açı ve yön bulma işleminin matematiksel açılımı aşağıda belirtilmiştir.

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \tan^{-1}((L(x, y + 1) - L(x, y - 1)) / ((L(x + 1, y) - L(x - 1, y)))$$

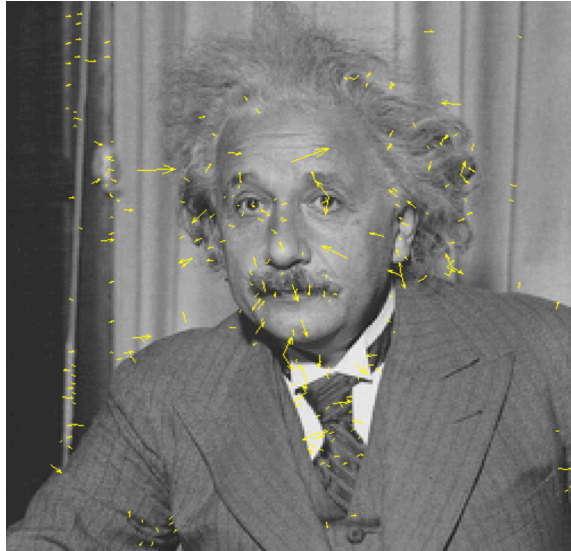
Aşağıdaki formülde L, uygulanan ölçekteki resmi temsil etmektedir. 'm' yön ağırlığını, 'θ' ise açığı temsil etmektedir. Daha sonra elde edilen açı değerleri 10 derecelik grupta ile 36 ayrı sınıfa ayrılarak Şekil 7.'deki gibi bir histogram oluşturulmaktadır.



Şekil 8. Gradyan Yön Ağırlıklarının Histogram Gösterimi

Kaynak: Sinha, U. (2018). *SIFT: Theory and Practice*

Elde edilmiş histogramda eğer %80’i aşan bir değer var ise; bu yeni bir önemli nokta(interest point) olarak kaydedilir. Bu noktanın konumu aynı olur fakat gradyan yön ağırlı farklı olarak listeye eklenir.



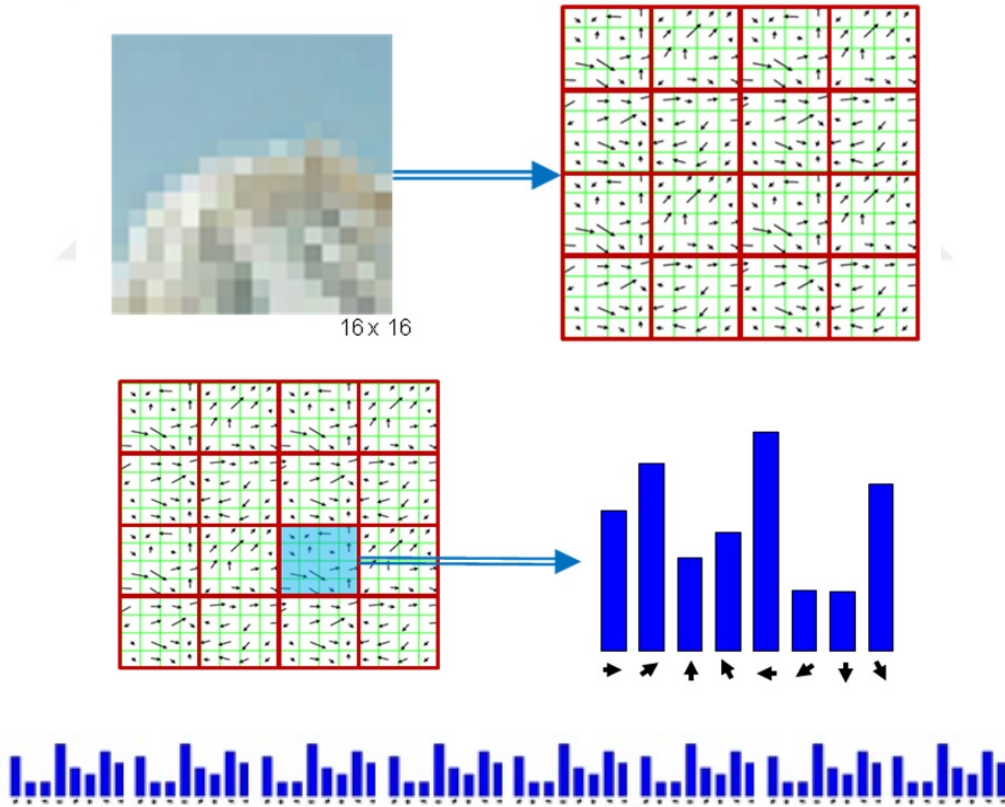
Şekil 9. Yön ağırlıklarının resim üzerinde gösterilmesi

Kaynak: F. Estrada, A.J., D. Fleet (2004) *Local Features Tutorial*

3.8 Önemli Noktaların İmzasının Çıkartılması

Bu işlem için öncelikle önemli noktanın merkezinde tutulduğu 16x16'lık bir çerçeve belirlenir. Daha sonra bu çerçeve içerisindeki tüm pixel'lerin açısı ve yoğunluğu yukarıdaki metotta anlatıldığı gibi çıkartılır. Daha sonra bu 16x16'lık çerçeve kendi içerisinde 4 adet 4x4'lük çerçevelere bölünür ve her bir çerçeve için yukarıda belirtilen metod ile histogram elde edilir. Bu sefer açılar 0-45; 45-90.. dereceler olmak üzere 8 ayrı sınıfta gruplandırılır. Elde edilen tüm histogramlar bir araya getirildiğinde (8 sınıf) x (16 blok) = 128'lik bir vektör elde edilir. Bu 128'lik vektör; önemli noktanın eşleşmesi esnasında kullanılmak üzere imza olarak kabul edilir.

Yukarıda anlatılan yöntemin temsili gösteri Şekil 8.'deki gibidir.



Şekil 10. Önemli noktaların imzalarının çıkartılması

Kaynak: London, I. *Image Classification in Python with SIFT Features*. 2016

3.9 Parlaklıktan bağımsızlığın eklenmesi

SIFT algoritması şu ana kadar yapılan işlemler ile büyüklük ve açı değişimlerinden bağımsız hale getirilmişti, elde edilmiş 128'lik vektör üzerinde normalizasyon işlemi uygulanarak algoritmanın parlaklıktan bağımsızlık özelliğide elde edilmiştir. [1]

3.10 SIFT Algoritmasının tercih sebebi

SIFT algoritması bu başlıkta belirtildiği gibi 1999 senesinde geliştirilmiş bir algoritmadır. [1] SIFT algoritmasından sonra birçok SIFT tabanlı varyasyonlar ve farklı algoritmalar geliştirilmiştir. Bu çalışmada SIFT algoritmasının seçilmesinin sebebi, yayınlandığı tarihten itibaren bir çok uygulamada kullanılması ve üzerinde geliştirilmiş birçok kütüphane geliştirilmiş olmasıdır. Tez kapsamı içerisinde geliştirilen mobil uygulama sürecinde Resim Tanıma Sistemi'nde kullanılan algoritmanın olası bir hata durumunda alternatif kütüphanelerinin de bulunması ve algoritma implementasyonunu geliştiricilerin hataları giderebilecek bir topluluğa sahip olması göz önünde bulundurulmuştur. Tez kapsamındaki eser eşleme sürecinde SIFT algoritmasının mobil uygulama gereksinimlerini sağlaması yeterli görülmüştür.

4. MOBİL UYGULAMA İÇİN SIFT ALGORİTMASI

Bu bölümde SIFT algoritmasının bir web-servisine çevrilmesi ve performansına yönelik bazı optimizasyonlar ile eşleşme süresini mobil uygulama düzeyinde kullanılması anlatılmaktadır.

4.1 Konum Tabanlı Optimizasyon

Uygulama içerisinde eşleşme yapılacak eserlerin tümü konum-bazlı eserler olmasından dolayı öncelikle mobil uygulamadan gelen enlem ve boylam bilgilerine göre filtreleme uygulanmıştır. Bu filtreleme de aranacak eserlerin adedini azaltarak daha hızlı bir sonuç alınması hedeflenmiştir. Bu filtreleme sayesinde eserin 30 metre civarındaki tüm eserler tesbit edilerek arama algoritması sadece bu eserler üzerinde uygulanmıştır. 30 metrelik bir alanın filtrelenmesi mevcut mobil cihazlardaki GPS sapmalarından dolayı tercih edilmiştir. [9] Herbir resime ait enlem ve boylam bilgileri resimlerin veritabanına kaydedilirken tuttulan id'lerin sonuna eklenmiş. Böylece resim tanıma sisteminin kolaylıkla bu filtremeyi kullanması sağlanmıştır.

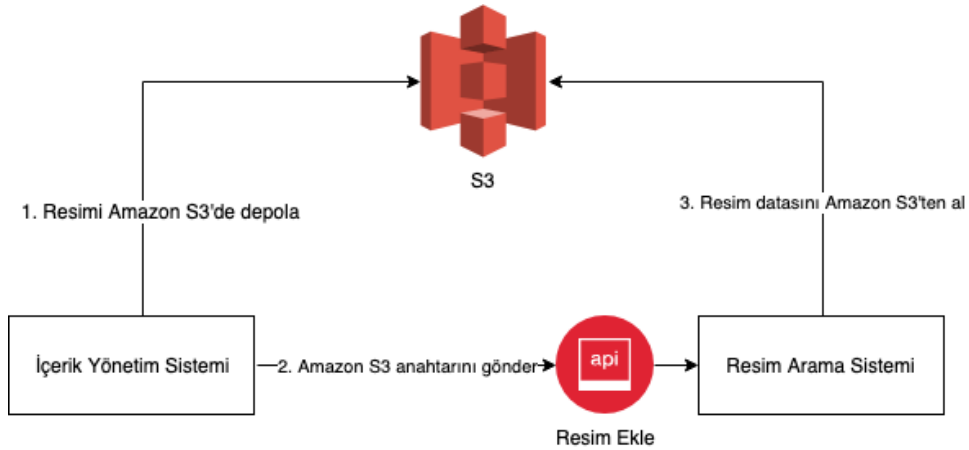
4.2 Apache Lucene Arama Servisi

Apache Lucene, Java yazılım dili ile geliştirilmiş ve günümüz modern uygulamalarda sıklıkla kullanılan açık kaynak kodlu bir arama altyapısıdır. [10] Lucene altyapısını kullanan uygulamalar arasında, Twitter, Apple ve Wikipedia en meşhurlardır [11]. Apache Lucene üzerine yazılmış en bilinen servis Solr; yüksek performanslı arama servisi olarak bilinmektedir [12]. Bu tez kapsamında SIFT algoritmasından üretilen 128'lik vektörleri Apache Lucene üzerinde tutulması tercih edilmiştir. Bu tercihin en önemli nedeni; resim eşleme esnasında aranacak önemli nokta imzalarının, Apache Lucene içerisinde hazır bulunan öklit mesafe ölçümü(Euclied Distance) özelliğinden faydalanılarak elde edilmesidir [13]. Bu işlem Apache Lucene tarafından optimize ve yüksek performanslı bir şekilde sağlanmaktadır. Ayrıca yine Apache Lucene üzerinde konum tabanlı arama yapılabilmektedir. Aranacak eserlerin konum tabanlı filtrelenmesinde bu özellikten faydalanılmıştır [13]. Apache Lucene kullanılmasının bir diğer tercih sebebi ise; resim tanıma sistemindeki verilerin artması durumunda, arama altyapısı tarafından sağlanan dağıtım(distribution) ve

kopyalama(replication) hizmetlerinden faydalanarak resim tanıma sisteminde oluşabilecek yükün giderilmesini sağlanmasıdır [14].

4.3 Resimlerin Depolanması

Uygulama içerisinde gerek Resim Tanıma Sistemi tarafında gerekse Mobil Uygulama tarafında bir çok resim datası depolanacaktır. Bu resim datalarına kolaylıkla erişim ve maliyeti düşürmek amaçlı bulut depolama hizmeti olarak bilinen Amazon S3 ürünü tercih edilmiştir [15]. Uygulama içerisindeki tüm resim datası akışları Amazon S3 üzerindeki klasörlere düzenli olarak düzenlenmiştir. İçerik Yönetim Sistemi tarafından yeni bir eser eklendiğinde; eserin resimleri öncelikle Amazon S3 içerisindeki bir klasöre eklenmektedir ve Amazon S3 üzerindeki anahtar ismi Resim Tanıma Sistemine iletilmiştir. Mobil uygulama üzerinden bir resim aratılacağı zaman aynı şekilde resim ilk önce Amazon S3 üzerindeki bir klasöre yüklenerek, buradaki anahtar ismi Resim Tanıma Sistemine iletilmiştir. Resim datalarının hem İçerik Yönetim Sistemi hem de Resim Tanıma Sisteminde tutularak oluşacak veri maliyetini asgari seviyeye indirmek için böyle bir akış tercih edilmiştir. Ayrıca ileride Amazon S3'ün yedekleme mekanizmasından da faydalanılması düşünülmüştür. Yukarıda anlatılan akış Şekil 9.'da gösterilmektedir.



Şekil 11. Amazon S3 üzerinden resim datası akışı

4.4 SIFT Algoritmasının Rest Servise Çevrilmesi

REST(Representational State Transfer); web servislerinde kullanılan bir mimari tasarım tipidir [16]. Kullanımındaki kolaylıktan dolayı günümüz bir çok uygulamada tercih edilmektedir. Bu tezde SIFT algoritmasının internet üzerinden erişilebilmesi için JAVA dilinde kullanılan ve açık kaynak kodlu olan SPRING BOOT altyapısı kullanılmaktadır [17]. Bu altyapı bir JAVA uygulamasının kolaylıkla REST servislerine çevrilmesine olanak tanımaktadır. Tez kapsamında 6 adet REST servisi oluşturulmuştur.

4.4.1 Resim Ekleme Servisi

Resim tanıma sisteminin yüklendiği server'ın 'index/s3' adresine POST tipinde istek yapılarak erişilir. Gönderilen POST istediğinin içeriğinde(request body) aşağıdaki JSON formatında, resim tanıma sistemine eklenmek istenen resimlerin Amazon S3 anahtarları gönderilir. (Bknz 4.3. madde)

```
{  
  "ids": [  
    "feb0f416-b399-4f40-9bed-b9737c0bb7db_41.01714_29.02744.jpg",  
    "fee063ad-7a89-43b9-899d-6432620bff03_41.01714_29.02744.jpg",  
  ]  
}
```

Gönderilen tüm resimlerin SIFT bilgilerinin çıkartılması ve arama sistemine kaydedilmesi vakit alacağına işlem süresi tamamlanmadan kullanıcıya doğrudan 201 kabul edildi istek değeri döndürülür. Servis tarafından dönen değeri beklemeden gerçekleşen bu tip işlemler asenkron servisler olarak da bilinmektedir [18].

4.4.2- Resim Arama Servisi

Resim tanıma sisteminin yüklendiği server'ın 'search/s3' adresine POST tipinde istek yapılarak erişilir. Gönderilen POST istediğinin içeriğinde(request body) aşağıdaki JSON formatında, Resim Tanıma Sisteminde aramak istenilen resimin Amazon S3 anahtarı ve aranan eserin enlem ve boylamı gönderilir.


```
{
  "imageId": "016f085e-8a5f-488cb380-56f958f08715_41.01714_29.02744.jpg",
  "location": {
    "latitude": 41.01714,
    "longitude": 29.02744
  }
}
```

Resim Tanıma Sistemi kendisine verilen bilgiler ile kendi veritabanındaki bilgileri eşleştirmeye çalışır ve sonucu aşağıdaki JSON formatında döndürür.

```
[
  {
    "imageId": "016f085e-8a5f-488c-b380-56f958f08715_41.01714_29.02744.jpg",
    "score": 818
  },
  {
    "imageId": "01121f085e-8a5f-488c-b380-56f958f0871_41.01714_29.02744.jpg",
    "score": 714
  },
  {
    "imageId": "16f08141e-8a5f-488c-380-56f958f08715_41.01714_29.02744.jpg",
    "score": 410
  }
]
```

Dönüş yapılan JSON nesnesinde ‘imageId’ eşleşen resimlerin Amazon S3 üzerindeki anahtarlarının değeridir. ‘score’ ise aranan resimin kaç önemli nokta(interest point) ile eşleştiğini belirtir. Mobil uygulama; gönderilen skora göre sıralama yaparak kullanıcıya eşleşen bilgileri göstermektedir.

4.4.3 - Resim Silme Servisi

Resim tanıma sisteminin yüklendiği server'ın 'index/s3' adresine DEL tipinde istek yapılarak erişilir. Gönderilen DEL istediğinin içeriğinde(request body) aşağıdaki JSON formatında, resim tanıma sisteminden silinmek istenen resimlerin Amazon S3 üzerindeki anahtarları bulunmaktadır.

```
{
  "ids": [
    "feb0f416-b399-4f40-9bed-b9737c0bb7db_41.01714_29.02744.jpg",
    "fee063ad-7a89-43b9-899d-6432620bff03_41.01714_29.02744.jpg",
  ]
}
```

Burada da Resim Ekleme Servisinde olduğu gibi karşı tarafa doğrudan 201 istek kabul edildi cevabı döndürülerek, karşı tarafın beklemeden işlemlerine devam etmesi sağlanmıştır.

4.4.4 - Resim Adedi Servisi

Resim Tanıma Sisteminin yüklendiği server'ın 'index/count' adresine GET tipinde istek yapılarak erişilir. Döndürdüğü değer sayı(integer) tipinde bir değerdir. Resim Tanıma Sisteminde kaç eser olduğunu belirtir. Böylece İçerik Yönetim Sisteminde bulunan eserler ile Resim Tanıma Sisteminde bulunan eserlerin birlerine eş gidip gitmedikleri kontrol edilir.

4.4.5 - Resim Kontrol Servisi

Resim tanıma sisteminin yüklendiği server'ın 'index/check/{Amazon_S3_Anahtarı}' adresine GET tipinde istek yapılarak erişilir. Parametre olarak sistemde olup olmadığı kontrol edilmek istenen resim anahtarı gönderilir. Bu servis İçerik Yönetim Sistemindeki herhangi bir güncelleme durumunda aynı güncellenenin Resim Tanıma Sisteminde de uygulanmasında kullanılmaktadır. İçerik Yönetim Sistemindeki bir eser güncellendiğinde; eklenen resimlerin tümü bu

servis vasıtası ile Resim Tanıma Sisteminde mevcut mu diye kontrol edilir ve mevcut olmayan tüm eserler için 'Resim Ekleme Servisi' çalıştırılır.

4.4.6 - Resim Tanıma Durum Servisi

Resim tanıma sisteminin yüklendiği server'ın 'actuator/health' adresine GET tipinde istek yapılarak erişilir. Döndürdüğü değer 200 Başarılı veya 400 Başarısız değeridir. İçerik Yönetim Sistemi, Resim Tanıma Servisini kullanmadan önce bu servis ile Resim Tanıma Sisteminin durumunu kontrol eder. Eğer Resim Tanıma Servisinde herhangi bir hata mevcut ise kullanıcıya eserlerin Resim Tanıma Sistemine kayıt olmadığını belirtir.



5. İÇERİK YÖNETİM SİSTEMİ

İçerik Yönetim Sistemi, çok sayıda insanın internet üzerinden erişerek içerik üretmesini sağlayan sistemdir [19]. Tez kapsamında ise içerisinde geliştirilen mobil uygulamanın içeriklerinin yönetildiği sistemdir. İçerik Yönetim Sistemi geliştirilirken aynı zamanda ileride akademik çalışmalarda kullanılmak üzere bir envanter yazılımı olması hedeflenmiştir. Mobil uygulama performansı nedeni ile veritabanı tipi olarak geleneksel ilişkisel veritabanı yerine döküman model tipinde ve açık kaynak kodlu MongoDB tercih edilmiştir [20]. MongoDB aynı zamanda NOSQL veritabanı olarakta bilinmektedir ve Google, CISCO, Facebook vb. bir çok geniş ölçekli firma tarafından da tercih edilmektedir [21]. İçerik Yönetim Sisteminin web servisleri için Amazon Lambda teknolojisi tercih edilmiştir [22]. Amazon Lamda, servislerin güvenliği, log işlemleri gibi birçok hizmeti içerisinde barındırmaktadır. Ayrıca bir server kiralamak yerine kullandıkça-öde hizmetleri ile mobil uygulamanın maliyetini düşürmüştür. İçerik Yönetim Sisteminin servisleri NodeJS yazılım dili yazılmış ve Serverless kütüphanesi kullanılarak Lamda üzerinde çalışmaya uygun hale getirilmiştir [23]. İçerik Yönetim Sisteminin arayüz tarafı ise PHP yazılım dili ile yazılmış ve Laravel alt yapısını kullanmıştır [24]. Laravel kütüphanesi; bir internet sitesi üzerindeki akışları kolaylaştırmaya yarayan bir kütüphanedir ve aynı zaman MVC mimarisi olarak da bilinmektedir [25]. İçerik Yönetim Sisteminin tek bir mimari ile yazılmayıp servis katmanı ile ön yüz katmanının birbirinden ayrı tutulmasının sebebi; mobil uygulama tarafından gelen yoğun isteklerin İçerik Yönetim Sisteminde yavaşlığa sebep olmamasıdır.

5.1 Rol Yönetimi

İçerik Yönetim Sistemi içerisinde uzman, gönüllü ve yönetici olmak üzere üç tip rol belirlenmiştir ve kullanıcılara verilen roller ile bazı işlemlerin kısıtlanması sağlanmıştır. Buna göre 'gönüllü' rolündeki kullanıcılar İçerik Yönetim Sistemine bilgi girişi yapabilirler fakat girilen bilginin durumunu 'yayına hazır' yapamazlar.

'Uzman' rolüne sahip kullanıcılar sisteme doğrudan bilgi girişi yapabilir ve durumunu 'yayına hazır' hale getirebilir. Ayrıca bu role sahip kullanıcılar, gönüllüler tarafından girilen bilgilerin kontrolünü sağladıktan sonra eserin durumunu 'yayına

hazır' yapabilirler. Resim Tanıma Sistemi tarafından eşleşen bir resim, İçerik Yönetim Sisteminde bulunan karşılığında eğer durumu 'yayına hazır' değil ise; kullanıcıya eser bulunamadı sonucu gösterilerek; kullanıcıya sadece uzmanlar tarafından onaylanmış eserlerin iletilmesi sağlanmıştır.

'Yönetici' rolündeki kullanıcılar ise sistem üzerinde kullanıcı oluşturma, kullanıcılara rol verebilme yetkisine sahiptirler.

5.2 Eserlerin Tanımlanması

İçerik Yönetim Sistemi içerisinde bulunacak eserler: kitabe, hat eserleri ve şahideler olmak üzere üç kategoride ele alınıp; eserler hakkında aşağıdaki bilgiler tespit edilmiştir.

5.2.1 Kitabe

Kitabe bir tarihi eserin; yapılış tarihi, yapılış nedeni, yaptıran kişi bilgilerini şiirsel bir dil ile taş veya mermer üzerine işlenmesidir. Genellikle dinî, sivil ve askerî binaların belirli yerlerine özenle işlenen kitâbe, verdiği bilgilerle ve yapının estetiğini tamamlayan dekoratif bir unsur olmasıyla önem taşıyan bir mimari elemandır [26]. Kitabeler için İçerik Yönetim Sisteminde aşağıdaki alanlar belirlenmiştir.

- Kitabe resimleri: Eşleşme algoritmasında kullanılmak üzere 2048 pixel 72 dpi seviyesinde kitabenin resimleri içerir.
- Bulunduğu bina türü: Kitabenin ne tür bir binada bulunduğu. Örneğin: cami, çeşme, medrese
- Kitabenin türü: Kitabenin türü hakkında bilgileri içerir. Örneğin: İmaret kitabesi, çeşme kitabesi
- Kitabenin edebi türü: Kitabenin hangi tür edebi türe ait olduğu bilgileri içerir. Örneğin: Nesir, beyit
- Kitabenin sanatçısı: Kitabenin hangi sanatçı tarafından yapıldığı hakkında bilgileri içerir.
- Kitabenin dili: Kitabenin hangi dilde yazılmış olduğu bilgileri içerir. Örneğin: Farsça, Arapça, Türkçe

- Kitabenin yazı tipi: Kitabede hangi tip yazı tipinin kullanıldığı bilgilerini içerir.
Örneğin: Talik, Rika, Sülüs
- Kitabenin vezini: Kitabede hangi edebi vezin kullanıldığı hakkında bilgileri içerir.
Örneğin: Failatün failatün failün
- Kitabenin içeriği: Kitabe üzerindeki metni içerir.
- Kitabedeki ebced notu: Kitabe üzerindeki ebced hesabı hakkındaki bilgileri içerir.
- Konumu: Resim eşleşme esnasında kullanılmak üzere eserin enlem ve boylam bilgilerini içerir.



Şekil 12. Sünbül Efendi Camii'nde bir kitabe örneği

5.2.2 Hat Eserleri

“Yazmak, çizmek; kazmak; alâmet koymak” anlamlarındaki Arapça *ḥatt* masdarından türeyen ve “yazı, çizgi; çığır, yol” gibi mânalara gelen hat kelimesi (çoğulu *ḥuṭṭ* ve *aḥṭât*), terim olarak “Arap yazısını estetik ölçülere bağlı kalıp güzel bir şekilde yazma sanatı (hüsnü’l-hat, hüsn-i hat)” anlamında kullanılmıştır. Osmanlı’da günlük yazışmalar için ağzı 1 milimetreyi geçmeyen kamaş kalemle yazılan rik’a hattı kullanılmıştır. Ta’lik, Sülüs hatları ise daha çok levhalar üzerinde kullanılmagelmiştir [27]. Hat eserleri; camii, tekke, türbe gibi bir çok alanda yaygın şekilde levha şeklinde duvarlara asılmıştır. Tez kapsamında içeriği Arapça ve Osmanlıca olan bu levhaların okunması sağlanılmaya çalışılmıştır. Sadece levha üzerinde değil; çini üzerinde yazılan her türlü hat eseri İçerik Yönetim Sistemine eklenmesi hedeflenmektedir.

Hat eserleri için İçerik Yönetim Sisteminde belirlenen alanlar aşağıdaki gibidir:

- Hat resimleri: Eşleşme algoritmasında kullanılmak üzere 2048 pixel 72 dpi seviyesinde hat eserinin resimlerini içerir.
- Hatın sanatçısı: Hat eserini yazan sanatçının bilgilerini içerir.
- Hattın içeriği: Hat eseri üzerindeki metni içerir.
- Hattın yazı tipi: Hat eserinde hangi tip yazı tipinin kullanıldığı bilgilerini içerir.
Örneğin: Talik, Rika, Sülüs
- Hattın sanat dalı: Hattın hangi sanat dalına ait olduğu bilgilerini içerir. Örneğin. Divani, Tuğra
- Konumu: Resim eşleşme esnasında kullanılmak üzere eserin enlem ve boylam bilgilerini içerir.



Şekil 13. Mahmut Celaleddin Efendi'ye ait bir hat levhası

5.2.3 Şahide

Şahideler aynı zaman mezar taşı olarak da bilinmektedir. Bir mezarın baş kısmına dikilen taştan mamul işaretlerdir. Şahideler üzerindeki başlıklar, motifler, şekilleri oradaki mevtanın yaşadığı dönemi, mesleği, devlet bünyesindeki kademesi gibi bilgileri içerir [28].

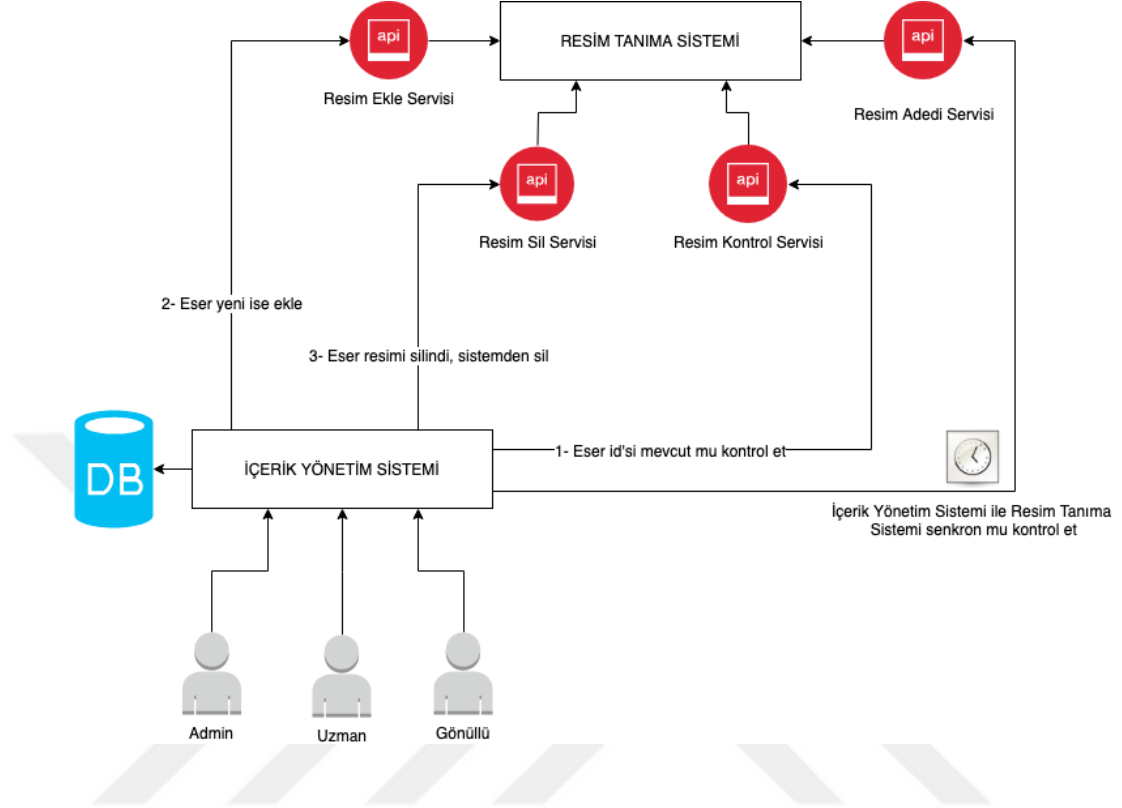
Şahideler için İçerik Yönetim Sisteminde aşağıdaki alanlar belirlenmiştir.

- Şahidenin resimleri: Eşleşme algoritmasında kullanılmak üzere 2048 pixel 72 dpi seviyesinde şahidenin resimlerini içerir.
- Adı: Şahide üzerinde belirtilen isimi içerir.
- Cinsiyeti: Şahide üzerinde belirtilen cinsiyeti içerir.
- Tarihi: Şahide üzerindeki tarihi içerir.
- Şahide metin hattatı: Şahide üzerinde işlenmiş hattın sanatçısı bilgilerini içerir.
- Şahide başlık tipi: Şahidenin başlığı hakkında bilgileri içerir. Örneğin: Katib, Fes
- Şahide tipi: Şahide tipi bilgilerini içerir. Örneğin: Masa Lahitli, Çerçeveli
- Şahide yazı tipi: Şahide üzerindeki hattın yazı tipi bilgilerini içerir. Örneğin: Talik, Rika, Sülüs.
- Şahide meslek bilgisi: Şahide üzerinde belirtilmiş meslek bilgilerini içerir.
- Şahide içeriği: Şahide üzerinde bulunan hattın metnin içerir.
- Şahide üstündeki süsleme notları: Şahide üzerindeki süsleme bilgilerini içerir.
- Konumu: Resim eşleşme esnasında kullanılmak üzere eserin enlem ve boylam bilgilerini içerir.



Şekil 14. Fatih Camii içerisindeki bir şahide örneği

6. İÇERİK YÖNETİM SİSTEMİ İLE RESİM TANIMA SİSTEMİ ENTEGRASYONU



Şekil 15. İçerik ekleme, silme ve güncelleme senaryosu temsili gösterim

İçerik Yönetim Sistemine yeni bir eser eklendiğinde veya bir eserin resimleri güncellendiği zaman Resim Tanıma Sistemine yeni resim eklenir veya güncelleme yapılır. Bu işlem için öncelikle Resim Kontrol Servisi çağırılarak resimin daha önce sistemde mevcut olup olmadığı kontrol edilir(1). Eğer resim ilk defa ekleniyorsa aşağıdaki formatta yeni bir anahtar oluşturulur.

mongodbId(13 karakter)_uuiid(8 karakter)_eserTipi_enlem_boylam.jpg

Resim Tanıma Sisteminden dönen eserlerin İçerik Yönetim Sistemindeki bilgiler ile doğru bir şekilde eşleştirilmesi için anahtarın baş kısmındaki mongoId kullanılmaktadır.

Resim Tanıma Sistemi; yeni bir resim eklendiği zaman gönderilen resmin anahtarını kaydederek arama sonuçlarında aynı ismi döndürecektir. Konum tabanlı filtrelemede kullanılmak üzere gerekli olan enlem ve boylam bilgileri resim isminin son

kısından kesilerek veritabanına kaydedilmektedir. Dosya ismindeki uuid(8 karakter) bir eserin birden fazla resmi olduėunda karışıklığa neden olunmaması için tutulmaktadır.

Resim Tanıma Sisteminde resim güncelleme durumunda karışıklığa sebep olacağından, güncellenen resimin eski kopyasının tamamen silinerek, yenisinin oluşturulması yöntemi tercih edilmiştir. Bu durumda İçerik Yönetim Sistemi, 'Resim Sil Servisi'ini çağırarak resmi siler ve yeni kopyasını gönderir.

İçerik Yönetim Sistemi ile Resim Tanıma Sistemi arasındaki senkronizasyon önemli olduğundan belli aralıklarla 'Resim Adedi Servisi' vasıtası ile İçerik Yönetim Sistemi ile Resim Tanıma Sistemindeki resim adedlerinin aynı olup olmadığı kontrol edilerek; İçerik Yönetim Sistemi'nin admin rolündeki kullanıcılara rapor olarak dönüş yapılmaktadır.

7. İÇERİK YÖNETİM SİSTEMİ İLE MOBİL UYGULAMA ENTEGRASYONU

Mobil uygulama üzerinden bir eser aratıldığında resim datası ve çekildiği enlem ve boylam bilgileri öncelikle Resim Tanıma Sistemine Base64 formatına dönüştürülerek gönderilir. Base64 formatı internet üzerinden dosya bilgilerinin gönderilmesi için oluşturulan bir protokoldür [29]. Resim Tanıma Sistemi'ne gönderilen arama isteğinin sonucundan eşleşen resimlerin bilgileri aşağıdaki JSON formatında döndürülür.

```
[
  {
    "imageId": "016f-8a5f-488c-b380-56f9_inscriptions_41.01714_29.02744.jpg",
    "score": 818
  },
  {
    "imageId": "0115e-8a5f-488c-b380-56f951_tomstones_41.01714_29.02744.jpg",
    "score": 714
  },
]
```

Resim Tanıma Sisteminden dönen değerler daha sonra İçerik Yönetim Sistemine iletilir. Sonuçlar; İçerik Yönetim Sistemi tarafından parçalanmaya başlar. Öncelikle eserTipi parametresinden hangi eser tipine ait veritabanı tablosunda(collection) arama yapılacağı belirlenir, daha sonra mongoDbId(13 karakter) eser anahtarı ile eserin bilgilerini bulunarak Resim Tanıma Sistemine geri döndürülür. Böylelikle Resim Tanıma Sisteminin, Mobil Uygulamaya eşleşen resim ile beraber İçerik Yönetim Sistemindeki bilgileri de döndürülmesi sağlanır.

8. MOBİL UYGULAMA VE TASARIM ÖRNEKLERİ

Mobil uygulama, eserlerin telefon kamerası ile çekilerek aratılmasını sağlamaktadır. Mobil uygulamaya giriş için kullanıcının öncelikle kayıt olması beklenmektedir (Ek-1). Ayrıca kullanıcıya sosyal medya üyelikleri ile kayıt olmasına olanak sağlanarak, uygulamaya bir sonraki girişince kolaylıkla erişebilmesi sağlanmıştır. Bu işlem aynı zaman Sosyal Medya ile Kayıt(Social Register) olarak da bir çok mobil uygulama tarafından kullanılmaktadır [30]. Daha sonra kullanıcı ana-ekrandan uygulama tarafından gösterilen eser detaylarına erişmesine olanak sağlanmıştır (Ek-4). Kullanıcı eser ara seçeceğine bastığında, uygulamanın yüklü olduğu işletim sisteminin sabit fotoğraf çekme ekranı gelmektedir (Ek-2). Fotoğraf çekimi esnasında kullanıcıdan konum bilgilerinin kullanılması için izin isteğinde bulunulur. Kullanıcı bu izini vermez ise; fotoğraf özelliğini kullanamamaktadır. Kullanıcının fotoğraf çekimini tamamlamasından sonra eşleşen resimler ekranı kullanıcıya gösterilmektedir(Ek-3). Ayrıca mobil uygulamaya bu tez kapsamına dahil edilmemiş, rota özelliği ve soru özelliği bulunmaktadır.

Aynı sanatçının eserleri bir araya getirilerek bir rota oluşturulabilir, kullanıcı bu rotayı takip edebilir. Bir eser eşleştiğinde eğer o eserle ilişkili bir soru mevcut ise; kullanıcı o soruyu bilerek puan kazanabilir gibi ek özellikler mobil uygulamaya eklenmiştir. Bütün bu bilgiler kullanıcının fotoğraf çekimi ardından gelen eser detayında gösterilmektedir. Eğer kullanıcı eserinde içerisinde bulunduğu bir rotayı takip etmeyi tercih ederse bu sefer Rotalar ekranına aktarılmaktadır (Ek-5). Burada kullanıcı, rotalara ait eserlerin listesi verilerek, rotadaki esere gitmesi ve oradaki eserin fotoğrafı çekmesi için yönlendirilir. Kullanıcı rotadaki bir eseri doğru şekilde bulduğunda uygulama tarafından kazandığı puan ve aynı sanatçıya ait diğer eserin konumu bildirilir (Ek-5). Böylece kullanıcının yaşadığı şehirdeki tarihi eserler ile etkileşim halinde olması hedeflenmektedir

Mobil Uygulama'dan alınmış tasarım görüntüleri Ekler bölümünde listelenmiştir.

9. TEST

Resim Tanıma Sisteminin kurulumu için 4 çekirdekli 8 GB ram'e sahip olan bir server tercih edilmiştir ve ortalama 20 sn. - 50 sn. arası bir arama süresinde doğru sonuç elde edilmektedir. Resim Tanıma Sistemini yüklediğimiz serverin konfigürasyonları kolaylıkla arttırılabilmektedir. Resim Tanıma Sistemi'nin verimliliğine göre buradaki değerler arttırılabilir ve bu süre azaltılabilir .Gerçek veriler ile test yapıldığında elde ettiğimiz sonuçlar aşağıdaki gibidir.



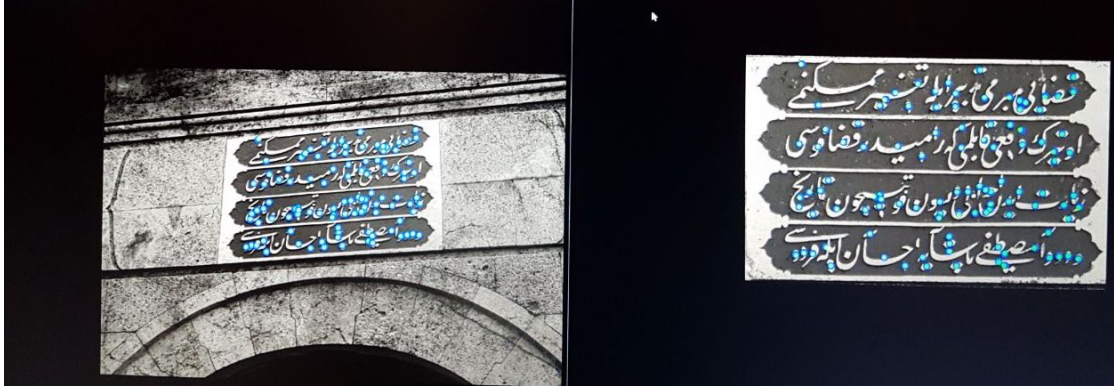
Şekil 16. Resim Tanıma Sisteminin gerçek veriler ile test edilmesi. Sağdaki aranan resim, solda doğru bulunan sonuç



Şekil 17. Resim Tanıma Sisteminin gerçek veriler ile test edilmesi 2. örnek
Sağdaki aranan resim, solda doğru bulunan sonuç



Şekil 18. Resim Tanıma Sisteminin gerçek veriler ile test edilmesi 3. örnek. Sağdaki aranan resim, solda doğru bulunan sonuç



Şekil 19. Resim Tanıma Sisteminin gerçek veriler ile test edilmesi 4. örnek. Sağdaki aranan resim, solda doğru bulunan sonuç

10. SONUÇ

Bu çalışmada SIFT algoritması kullanılarak Osmanlıca yazılı eserler üzerinde mobil uygulama üzerinden resim eşleşme yöntemi ile bilgilerine kolaylıkla erişim hedeflenmiştir.

Bu amaca yönelik öncelikle İçerik Yönetim Sistemi, Resim Tanıma Sistemi ve Mobil Uygulama detaylı bir şekilde tasarlanmış ve hayata geçirilmesi sağlanmıştır. SIFT algoritmasının tek başına süre bakımından böyle bir iş için yetersiz kalmasından dolayı lokasyon tabanlı filtre uygulanmış ve Apache Lucene gibi modern arama servisleri ile desteklenmiştir. SIFT algoritmasından üretilen imzaların geleneksel Öklid uzaklığı yöntemi ile eşleştirilmesi, mobil uygulamanın resim tanıma süresini çok uzatmaktadır ve bu kullanıcıya kötü bir deneyim olarak yansımaktadır. Bu eksiklik, bilgilerine erişilmek istenilen eserlerin konum tabanlı ve yeri değişmeyen, sabit birer varlık olmaları ve buldukları enlem ve boylam bilgileri ile filtrelenmesi ile aşılmıştır.

İçerik Yönetim Sistemi, ileride akademik çalışmalarda kullanılmak üzere tasarlanmış aynı zamanda Resim Tanıma Sistemi ile entegre edilerek, Mobil Uygulama üzerinden gelen sorgularıda cevaplaması sağlanmıştır. SIFT algoritması gibi bir resmin tamamı üzerinde arama yapılması yerine Karakter Tanıma Sistemi(OCR) yöntemi ile eserler üzerinden aratılması; daha uygun bir yöntemdir ve her bir eser için İçerik Yönetim Sisteminde bir kayıt tutulması zorunluluğunu ortadan kaldıracaktır. Fakat tez kapsamındaki eserlerin aynı zamanda devlet kapsamında Somut Olmayan Kültür Mirası olarak da nitelendirilmesi böyle bir envanter ve kayıt platforma ihtiyaç doğurmuştur. [31]

İleride İçerik Yönetim Sistemindeki eserlerin resim kayıtları ile içerik bilgileri; modern öğrenme algoritmalarında kullanılarak modellenmesi ve İçerik Yönetim Sisteminde henüz kaydı bulunmayan eserlerin üzerindeki bilgilerin de kullanıcıya iletilmesi daha uygun bir yöntem olacaktır.

Tez kapsamında küçük bir plot bölge seçilmiş ve İçerik Yönetim Sistemine girişleri yapılmıştır. Fakat Osmanlı Devletinin zaman içerisinde çok geniş bir coğrafyaya yayılmıştır. Hakim oldukları coğrafyada yapmış olduğu hayır eserlerinde kitabelerin kullanılması, Ortadoğu ve Balkan ülkelerinde günümüze ulaşmış Osmanlı şahidelerinin varlığı; Kudüs, Mekke, Medine gibi bir çok kutsal mekana hediye olarak gönderilmiş Osmanlıca ve Arapça hat levhalarının bulunması tez kapsamındaki hedefin çok geniş bir çerçevede ele alınmasına sebep olmaktadır. İçerik Yönetim Sisteminde kolaylıkla genişletilebilir olan MongoDB veritabanının tercih edilmesi bu nedenledir.

Ayrıca Mobil Uygulamanın yabancı dil desteği ile Türkiye ve Osmanlıca eserlerin bulunduğu diğer ülkeleri ziyaret ettiklerinde kullanılması hedeflenmektedir. Kültür ve Turizm Bakanlığı - Döner Sermaye İşletmesi Merkez Müdürlüğünün 2018 yılına ait yayınladığı ziyaretçi istatistiklerinde sadece Ayasofya Müzesi ve Topkapı Sarayı'na gelen ziyaretçi sayısı 5.000.000'u geçmektedir[32]. İleride yaşanılması muhtemel olan Mobil Uygulama tarafından gelen istekler yoğunluğu sistemin işleyişini olumsuz yöntem etkileme riskine sahiptir. Bu nedenle küçük ölçekli uygulamalar için de ileride büyüme potansiyeli sahip olan uygulamalar içinde, kullandıkça-öde fatura altyapısına sahip Bulut-tabanlı Amazon Lambda servisleri kullanılması doğru bir tercih olarak görülmüştür.

KAYNAKÇA

1. Lowe, D.G., Distinctive Image Features from Scale-Invariant Keypoints. 2004.
2. Janez Križaj, V.S., Nikola Pavešić, *Adaptation of SIFT Features for Robust Face Recognition*. 2010, Faculty of Electrical Engineering, University of Ljubljana.
3. Çevrimiçi, <http://aishack.in/tutorials/sift-scale-invariant-feature-transform-log-approximation> Erişim Tarihi: 01.05.2019
4. Çevrimiçi, <https://www.cs.toronto.edu/~jepson/csc2503/tutSIFT04.pdf> Erişim Tarihi: 03.05.2019
5. Fisher, R.B., *Subpixel Estimation*, in *Computer Vision*, K. Ikeuchi, Editor. 2016.
6. Çevrimiçi, <http://www.csce.uark.edu/~mqhuang/courses/5013/f2011/proj/sift.presentation.pdf> Erişim Tarihi: 03.05.2019
7. Çevrimiçi, https://en.wikipedia.org/wiki/Harris_Corner_Detector Erişim Tarihi: 05.05.2019
8. Çevrimiçi, <https://ianlondon.github.io/blog/how-to-sift-opencv/> Erişim Tarihi: 06.05.2019
9. Çevrimiçi, <https://www.gps.gov/systems/gps/performance/accuracy/> Erişim Tarihi: 07.05.2019
10. Çevrimiçi, <http://lucene.apache.org/> Erişim Tarihi: 09.05.2019
11. Çevrimiçi, <https://www.slideshare.net/lucidworks/search-at-twitter-presented-by-michael-busch-twitter> Erişim Tarihi: 10.05.2019
12. Çevrimiçi, <http://lucene.apache.org/solr/> Erişim Tarihi: 12.05.2019
13. Çevrimiçi, https://lucene.apache.org/solr/guide/7_7/spatial-search.html Erişim Tarihi: 12.05.2019
14. Çevrimiçi, https://lucene.apache.org/solr/guide/6_6/distributed-search-with-index-sharding.html Erişim Tarihi: 12.05.2019

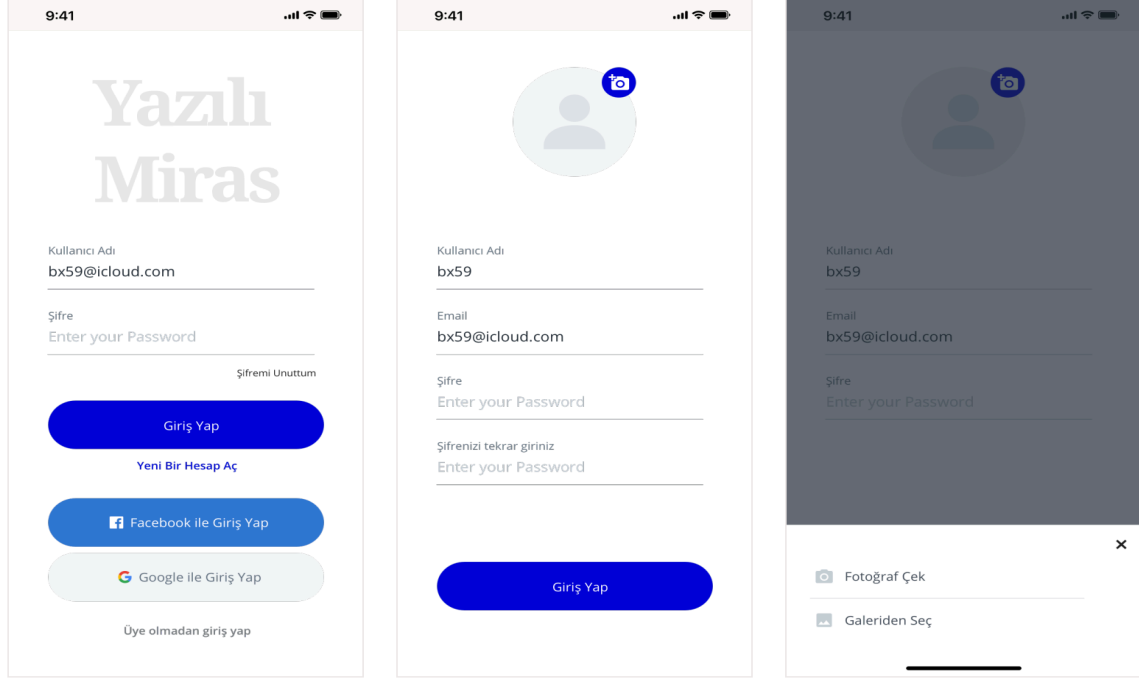
15. Çevrimiçi, <https://aws.amazon.com/tr/s3/> Erişim Tarihi: 13.05.2019
16. Çevrimiçi, https://en.wikipedia.org/wiki/Representational_state_transfer
Erişim Tarihi: 13.05.2019
17. Çevrimiçi, <https://spring.io/projects/spring-boot> Erişim Tarihi: 15.05.2019
18. Çevrimiçi, <http://restcookbook.com/Resources/asynchronous-operations/>
Erişim Tarihi: 15.05.2019
19. Çevrimiçi,
https://tr.wikipedia.org/wiki/%C4%B0%C3%A7erik_y%C3%B6netim_sistemi Erişim
Tarihi: 18.05.2019
20. Çevrimiçi, <https://www.mongodb.com/what-is-mongodb> Erişim Tarihi:
18.05.2019
21. Çevrimiçi, <https://www.mongodb.com/who-uses-mongodb> Erişim Tarihi:
18.05.2019
22. Çevrimiçi, <https://aws.amazon.com/tr/lambda/> Erişim Tarihi: 19.05.2019
23. Çevrimiçi, <https://serverless.com/> Erişim Tarihi: 20.05.2019
24. Çevrimiçi, <https://laravel.com/> Erişim Tarihi: 21.05.2019
25. Çevrimiçi,
<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> Erişim
Tarihi: 23.05.2019
26. Arpaşlan, A., *KİTÂBE - كتابه*, in *TDV İslâm Ansiklopedisi*. 2002, Türkiye
Diyanet Vakfı: Ankara.
27. Derman, M.U., *Hat - الخط*, in *TDV İslâm Ansiklopedisi*. 1997, Türkiye
Diyanet Vakfı: Ankara. p. 427-437.
28. Bozkurt, N., *Mezarlık*, in *TDV İslâm Ansiklopedisi*. 2004, Türkiye Diyanet
Vakfı: Ankara. p. 519-522.
29. Çevrimiçi, <https://en.wikipedia.org/wiki/Base64> Erişim Tarihi: 27.05.2019
30. Çevrimiçi, https://en.wikipedia.org/wiki/Social_login Erişim Tarihi:
27.05.2019

31. Çevrimiçi, <http://aregem.kulturturizm.gov.tr/TR-46095/somut-olmayan-kulturel-miras.html> Erişim Tarihi: 29.05.2019
32. Çevrimiçi, <http://www.dosim.gov.tr/assets/documents/2018.pdf> Erişim Tarihi: 30.05.2019.

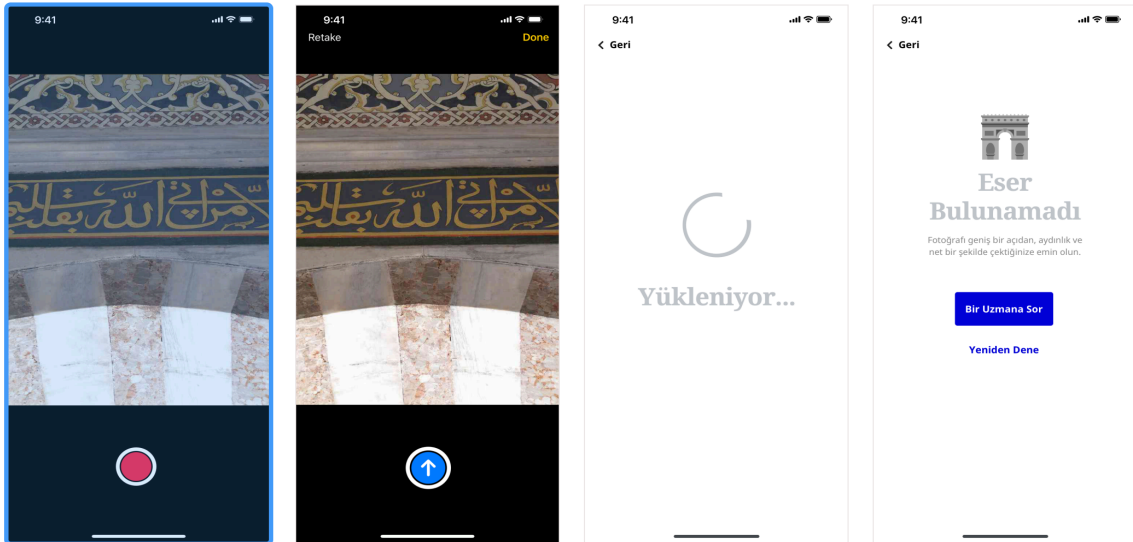


EKLER

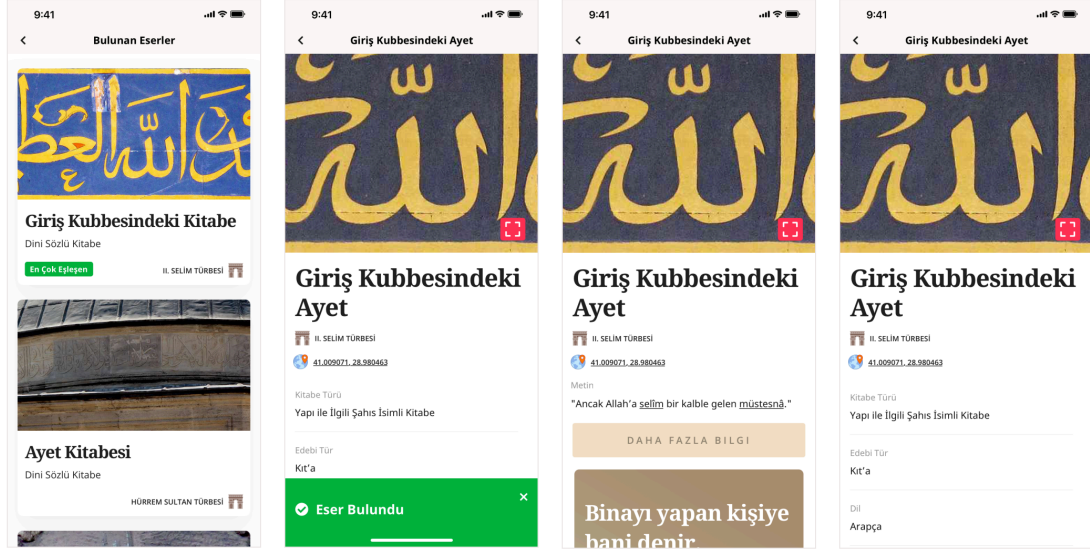
Ek-1: Mobil uygulama kayıt ve giriş ekranı



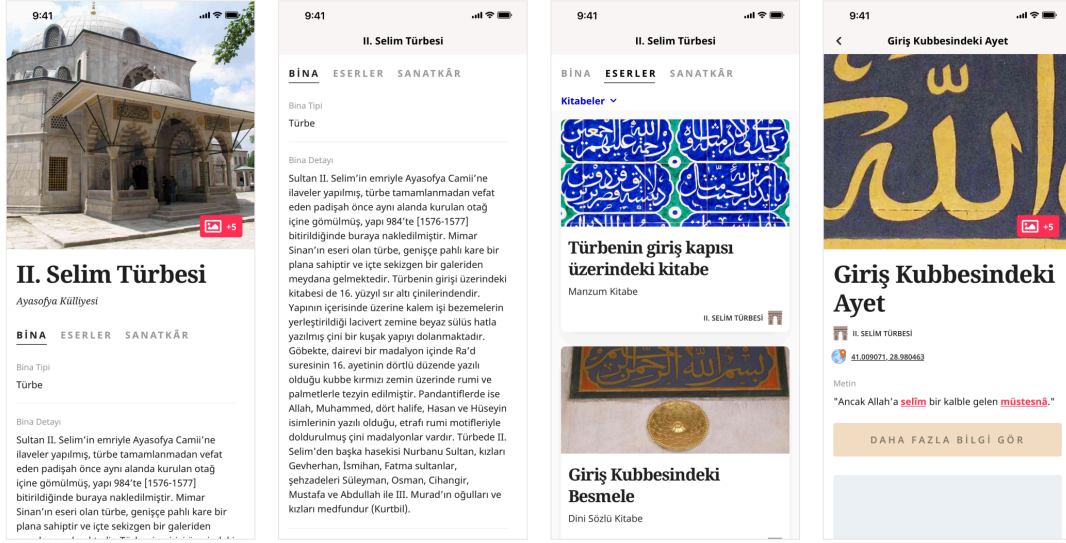
Ek-2: Mobil uygulama eser arama ekranı



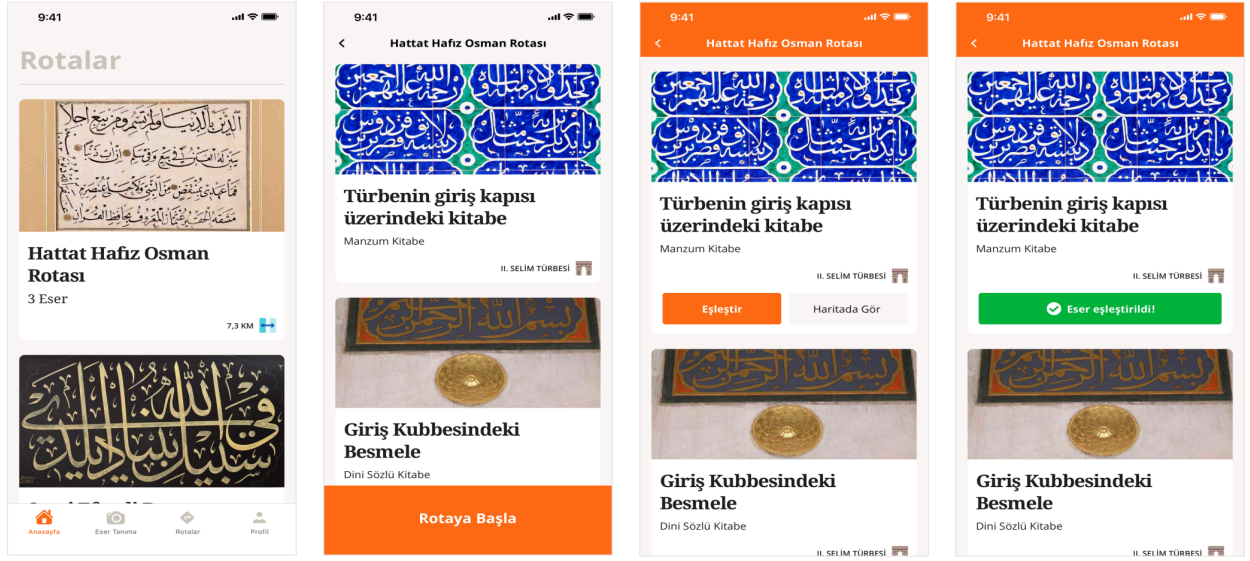
Ek-3 Mobil uygulama eşlenmiş eser ekranları



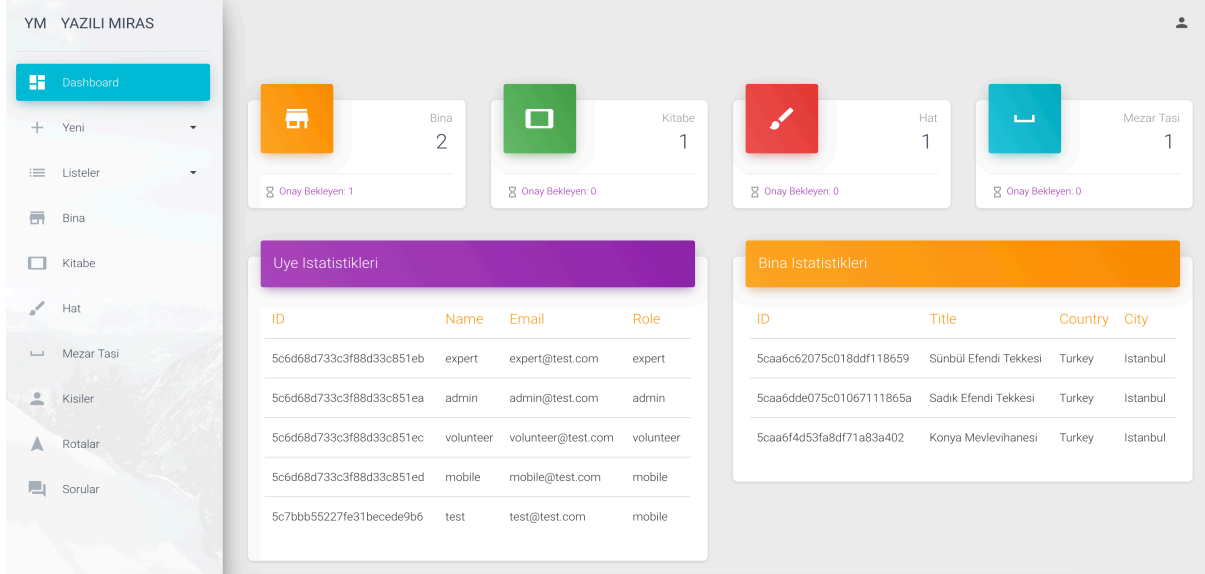
Ek-4 Mobil uygulama bina ve eser detayları ekranları



Ek-5 Mobil uygulama rota ekranları



Ek-6 İçerik Yönetim Sistemi Ana Ekran



Ek-7 İçerik Yönetim Sistemi Bina Ekle Ekranı

YM YAZILI MIRAS

- Dashboard
- Yeni
- Listeler
- Bina**
- Klabe
- Hat
- Mezar Tası
- Kisiler
- Rotalar
- Sorular

Bina Ekle

Baslık

Diğer Baslık

Resimler **Choose Files** No file chosen

Bina Tipi **BINA TÜRÜ SEÇİNİZ**

Açıklama

Sanatçı **SANATÇI SEÇİNİZ**

Bina Sebebi

Bina Tarihi

Tamir Tarihi

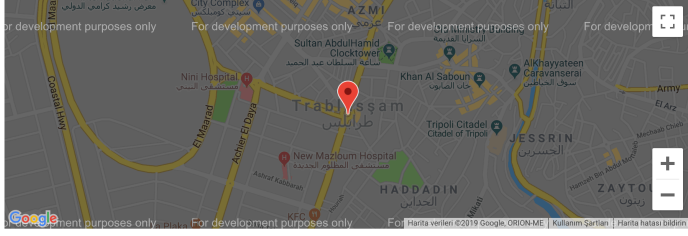
Ülke **Ülke Seçiniz**

Şehir

Bölge

Sokak

Konum



Konumu Onayla

latitude

longitude

Adres Notu

Kaynak

Durum Beklemede/Onaylandı

KAYDET

© 2019, Yazili Miras App.

Ek-8 İçerik Yönetim Sistemi Kitabe Ekle Ekranı

YM YAZILI MIRAS

Dashboard

Yeni

Listeler

Bina

Kitabe

Hat

Mezar Taşı

Kisiler

Rotalar

Sorular

Kitabe Ekle

Baslık

Resimler **Choose Files** No file chosen

Bina **Sünbül Efendi Tekkesi**

Kitabe Türü **Yapı ile ilgili şahıs isimli kitabe**

Edebi Tür **kit'a**

Sanatçı **Hamid Aytaç**

Dil **Arapça**

Yazı Tipi **Rika**

Vezin

İçerik

Unlicensed copy of the Froala Editor. Use it legally by purchasing a license.

Type something

0

Açıklama

Ebced Notu

Durumu

Kaynak

Location

Konu Onayla

latitude

longitude

Durum Beklemede/Onaylandı

KAYDET

© 2019, Yazılı Miras App.

Ek-9 İçerik Yönetim Sistemi Hat Ekle Ekranı

YM YAZILI MIRAS

- Dashboard
- Yeni
- Listeler
- Bina
- Kitabe
- Hat**
- Mezar Tasi
- Kisiler
- Rotalar
- Sorular

Hat Ekle

Baslik

Resimler **Choose Files** No file chosen

Bina **BINA SEÇİNİZ**

Sanatci **SANATCI SEÇİNİZ**

İcerik

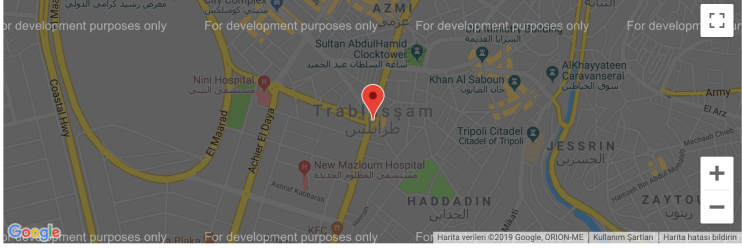
Unlicensed copy of the Froala Editor. Use it legally by purchasing a license.

Type something

Tipi

Sanat Dali

Location



Konumu Onayla Konumu boş bırakırsanız, yüklenen resimlerde eğer konum bilgisi varsa bu bilgileri otomatik doldurabiliriz.

latitude

longitude

Durum Beklemede/Onaylandı

KAYDET

© 2019, Yazili Miras App.

Ek-10 İçerik Yönetim Sistemi Şahide Ekle Ekranı

YM YAZILI MIRAS

Dashboard

Yeni

Listeler

Bina

Kitabe

Hat

Mezar Tasi

Kisiler

Rotalar

Sorular

Mezar Tasi Ekle

Baslik

Yeni Resim [Choose Files](#) No file chosen

Bina **BINA SEÇİNİZ**

Sanatci **SANATCI SEÇİNİZ**

Sair **SAIR SEÇİNİZ**

Hattat **HATTAT SEÇİNİZ**

Baslik Tipi **BASLIK TIPI SEÇİNİZ**

Mezar Tipi **MEZAR TIPI SEÇİNİZ**

Mezar Tasi Tipi **MEZAR TASI TIPI SEÇİNİZ**

Yazi Tipi **YAZI TIPI SEÇİNİZ**

Tarikat **TARIKAT SEÇİNİZ**

Tarikat Unvani **TARIKAT UNVANI SEÇİNİZ**

Meslek **MESLEK SEÇİNİZ**

Mensubiyet **MENSUBİYET SEÇİNİZ**

İçerik

Unlicensed copy of the Froala Editor. Use it legally by purchasing a license.

Type something

Aciklama

Susleme Notu

Adi

Cinsiyet

Akrabalik

Tarih

Tarih Turu **TARİH TURU SEÇİNİZ**

Kaynak

Konum

Konumu Onayla

latititude

longitude

Durum Beklemede/Onaylandı

KAYDET

© 2019, Yazili Miras App.

Ek-11 İçerik Yönetim Sistemi Kişi Ekle Ekranı

YM YAZILI MIRAS

Dashboard

+ Yeni

Listeler

Bina

Kitabe

Hat

Mezar Tasi

Kisiler

Rotalar

Sorular

Kisi Ekle

Adi

Resimler No file chosen


Hakkinda

Imza

Dogum Tarihi

Olum Tarihi

Location



Konumu Onayla

latitude

longitude

KAYDET

© 2019, Yazili Miras App.

Ek-12 İçerik Yönetim Sistemi Çoktan Seçmeli Soru Ekle Ekranı

YM YAZILI MIRAS

Dashboard

Yeni

Listeler

Bina

Kitabe

Hat

Mezar Tasi

Kisiler

Rotalar

Sorular

Soru Ekle

Soru Turu: ÇOKTAN SEÇMELİ

Soru

Cevaplar

Doğru seçeneği işaretleyiniz.

Doğru seçeneği işaretleyiniz.

Doğru seçeneği işaretleyiniz.

Doğru seçeneği işaretleyiniz.

Not: Cevap sayısı kadar cevap yazınız (Min:2-Max:4). Yazmadığınız cevaplar seçeneklerden silinecektir.

Sanat Eseri: HAT

ANA GİRİŞ ÜZERİNDEKİ H

KAYDET

© 2019, Yazili Miras App.

Ek-13 İçerik Yönetim Sistemi Doğru Yanlış Soru Ekle Ekranı

YM YAZILI MIRAS

Dashboard

Yeni

Listeler

Bina

Kitabe

Hat

Mezar Tasi

Kisiler

Rotalar

Sorular

Soru Ekle

Soru Turu: DOĞRU / YANLIŞ

Soru: Bani ne demektir?

Cevaplar

Binayı yapan kişi Doğru seçeneği işaretleyiniz.

Binada yaşayan kişi Doğru seçeneği işaretleyiniz.

Doğru seçeneği işaretleyiniz.

Doğru seçeneği işaretleyiniz.

Not: Cevap sayısı kadar cevap yazınız (Min:2-Max:4). Yazmadığınız cevaplar seçeneklerden silinecektir.

Sanat Eseri: HAT

ANA GİRİŞ ÜZERİNDEKİ H

KAYDET

© 2019, Yazili Miras App.

ÖZGEÇMİŞ

1990 yılında İstanbul Fatih’de doğdum. İlkokul, Ortaokul ve Lise eğitimimi İstanbul’da tamamladım. Kırklareli Pınarhisar Meslek Yüksekokulundan mezun olduğum sene Fulbright Bursu ile bir sene Northern Virginia Community College’de eğitim aldım. Döndüğümde Beykent Üniversitesi Yazılım Mühendisliği bölümünü tamamlayarak lisans eğitimimi tamamladım. 4 senedir TRT World bünyesinde yazılımcı olarak görev almaktayım.

Abdullah GÜRSOY