

**KARADENİZ TEKNİK ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**WINDOWS PROGRAMLAMA DERSİ DÖNEM PROJESİ ÖN RAPORU**  
**MOBİL CİHAZ ORTAMINDA ÇOKLU GÖRÜNTÜ BİRLEŞTİRME UYGULAMASI**

**Ders Sorumlusu:** Bekir DİZDAROĞLU

**Öğrenci Ad Soyad:** Osman Can AKSOY

**Öğrenci Numarası:** 394797

**2023**

## İÇİNDEKİLER

|  |   |
|--|---|
| 1. Kullanılması Planlanan Algoritmalar ve Yazılım Mimarileri .....   | 3 |
| 1.1. Önerilen Yöntemin Geliştirilmesi/Yeniden Kullanımı .....  | 4 |
| 1.1.1. Literatür Taramasına Göre Problemin Çözümünde Kullanılabilir Diğer Yöntemlerinin Açıklanması ve Aday Yöntemlerin Belirlenmesi ..... | 4 |
| 1.1.2. Problemin Çözümünde Kullanılacak Görüntü Veri Seti.....   | 5 |
| 1.1.3. Önerilen Yöntemin Test ve Geçerleme Aşamalarında Düzenlenecek Deneylerin ve Performans Ölçüm Yöntemlerinin Açıklanması .....        | 5 |
| 1.2. Yazılım Çözümü .....  | 6 |
| 1.2.1. Önerilen Yöntemin Test ve Kullanım Süreçlerine Ait Kullanım UseCase-Çizeneği6   |   |
| 1.2.2. Kullanılan Teknolojileri İçeren Yazılım Bileşen-Component-Çizeneği.....   | 7 |
| 2. Çözüm İçin Geliştirilen Yöntemler ve Bunların Özgünlüğe Katkısı .....   | 8 |
| 3. Proje Takvimi .....   | 8 |
| 4. Referanslar .....   | 8 |

## 1. Kullanılması Planlanan Algoritmalar ve Yazılım Mimarileri

Bu projede resim birleştirme işlemleri için OpenCV kütüphanesinde bulunan “Stitcher” sınıfını kullanımı planlanmıştır. Bu sınıf, farklı resimleri birleştirme işlemini gerçekleştiren önceden tanımlanmış algoritmalar içerir. Özellikle, resimlerin genel uyumluluğunu ve örtüşme bölgelerini algılayarak bunları tek bir büyük resim haline getirme işlevine sahiptir.

Bu algoritma, resimler arasındaki benzerlikleri ve dönüşümleri analiz eder. Ardından, resimleri hizalar ve birleştirme işlemini gerçekleştirir. Öncelikle resimlerin özelliklerini belirler ve ardından optimize edilmiş birleştirme sonucu elde edilir.

Proje C++ diline ait Qt framework'ü üzerinde geliştiriliyor ve OpenCV kütüphanesi entegre ediliyor. Bu entegrasyon, Qt'nin güçlü GUI bileşenleri ve QML (Qt Modeling Language) ile OpenCV'nin görüntü işleme yeteneklerini birleştirerek projeyi Android platformunda kullanılabilir hale getirme amacını taşıyor.

Qt, kullanıcı arayüzü geliştirmek için geniş imkanlar sunar. QML, özellikle dokunmatik cihazlar ve farklı ekran boyutları için esnek ve kolay bir arayüz geliştirme ortamı sunar. Bu yapı, projenin kullanıcı arayüzünü oluşturmak için tercih edilen bir yöntemdir.

OpenCV ise, resim işleme ve analizinde kapsamlı bir araç yelpazesi sunar. Bu kütüphane, resim seçme, işleme ve birleştirme gibi adımları içeren projenin resim işleme aşamalarını yönetmek için temel bir araç oluşturur.

Projedeki bu mimari yapı, QML kullanarak kullanıcı arayüzü geliştirmeyi ve ardından OpenCV'nin resim işleme yeteneklerini entegre etmeyi hedefler. Bu sayede Android platformunda çalışabilecek bir uygulama elde etme amaçlanır.

Bu yapı, hem kullanıcı dostu bir arayüz sağlama hem de resim işleme özelliklerini mobil cihazlarda kullanabilmek adına güçlü bir temel oluşturur.



### 1.1. Önerilen Yöntemin Geliştirilmesi/Yeniden Kullanımı

Projemizde kullanılan Qt, QML ve OpenCV entegrasyonu, Android platformunda etkili bir uygulama geliştirme temeli oluşturuyor. Bu entegrasyonun geliştirilmesi veya yeniden kullanılması için şu adımlar düşünülebilir:

- **Performans İyileştirmeleri:**

Farklı OpenCV ve Qt yapılandırmaları veya sürümleri kullanılarak performansı artırmak için çalışmalar yapılabilir. Büyük resimlerde veya karmaşık birleştirme işlemlerinde optimizasyonlar düşünülebilir.

- **Modüler Yapıların Geliştirilmesi:**

OpenCV işlemlerini modüller halinde düzenlemek, farklı projelerde veya bileşenlerde yeniden kullanılabilirliği artırabilir. Örneğin, resim seçme ve işleme adımlarını ayrı modüller olarak tasarlamak.

- **Taşınabilirlik ve Platform Destekliliği:**

Önerilen yöntemin sadece Android değil, farklı platformlarda da kullanılabilir olması önemlidir. iOS gibi diğer mobil platformlara taşınabilirlik üzerinde çalışılabilir.

- **Belgelendirme ve Topluluk Katılımı:**

Geliştirilen yöntemin belgelendirilmesi ve toplulukla paylaşılması, diğer geliştiricilerin benzer yapıları kullanmasını kolaylaştırabilir.

- **Esneklik ve Genişletilebilirlik:**

Qt ve OpenCV entegrasyonu sırasında, yapılan tercihlerin esneklik ve genişletilebilirlik üzerine odaklanması gelecekteki ihtiyaçlara uyum sağlamayı kolaylaştırır.

Bu adımlar, önerilen yöntemin daha güçlü ve kullanışlı hale getirilmesi için stratejik adımları içerir. Bu şekilde projenin geliştirilmesi aşamasında, daha verimli ve kullanılabilir bir yapı oluşturulabilir.

#### 1.1.1. Literatür Taramasına Göre Problemin Çözümünde Kullanılabilir Diğer Yöntemlerinin Açıklanması ve Aday Yöntemlerin Belirlenmesi

Görüntü birleştirme (image stitching) alanında yapılan literatür taraması, problemin çözümü için kullanılabilecek çeşitli yöntemleri ortaya koymaktadır. Bu bölümde, mevcut literatürde bulunan ve image stitching probleminin çözümüne katkı sağlayabilecek aday yöntemlere odaklanılacaktır.

Görüntüdeki benzer özellikleri tespit etmek ve kümelemek için kullanılan algoritmalar, özellikle kapsamlı manzara görüntülerinde daha etkili sonuçlar verebilir. Bu yöntemler, benzer özelliklere sahip görüntü bölgelerini tanımlamak için daha karmaşık algoritmalar kullanabilir.

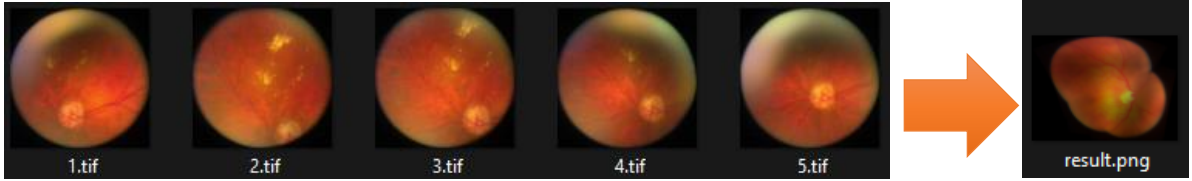
Görüntü birleştirme sürecinde kullanılan özellik eşleştirme algoritmaları için derin öğrenme modellerinin entegrasyonu, özellik çıkarma ve eşleştirme doğruluğunu artırabilir.

Özellikle önceden eğitilmiş derin öğrenme modelleri, genel görüntü eşleştirme performansını artırabilir. Görüntü birleştirme işleminin daha hızlı bir şekilde gerçekleştirilmesi için paralel işleme teknikleri ve GPU tabanlı çözümler araştırılabilir. Özellikle büyük görüntü verileri ile çalışıldığında performans artışı sağlayabilir.

SIFT ve ORB gibi geleneksel özellik eşleştirme algoritmalarının yanı sıra, daha hızlı çalışan ve düşük hesaplama maliyetine sahip yeni nesil algoritmaların incelenmesi gerekmektedir. Görüntüler arasındaki geçişleri daha doğal hale getirmek için gelişmiş kenar tespiti ve yumuşatma tekniklerinin kullanılması düşünülebilir. Bu, birleştirilen görüntülerin daha estetik bir bütün oluşturmaya yardımcı olabilir. Geleneksel RANSAC tabanlı homografi tahmini yerine, derin öğrenme modelleri kullanarak daha doğru ve hızlı homografi tahmini yapma stratejileri incelenebilir.

Yukarıda belirtilen bu aday yöntemler, literatür taraması sonucu belirlenen ve görüntü birleştirme probleminin çözümünde potansiyel katkı sağlayabilecek stratejilerdir. Her bir yöntemin avantajları ve dezavantajları, projenin spesifik gereksinimleri doğrultusunda değerlendirilerek uygun olanlar seçilebilir.

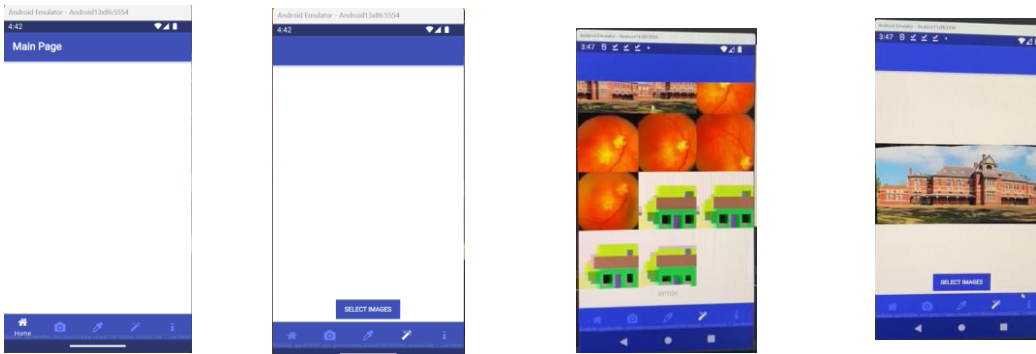
### 1.1.2. Problemin Çözümünde Kullanılacak Görüntü Veri Seti



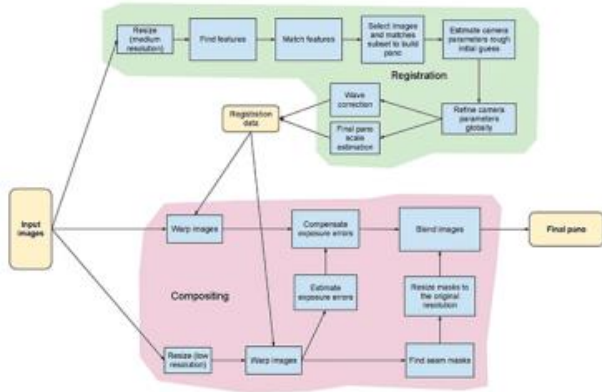
Kullanılan veri seti içerisinde çeşitli formatlarda göz retina görüntüleri yer almaktadır. Bu görüntüler ve bu görüntülerin başarılı eşleştirme yapılmış versiyonları veri seti içerisinde yer almaktadır.

### 1.1.3. Önerilen Yöntemin Test ve Geçerleme Aşamalarında Düzenlenecek Deneylerin ve Performans Ölçüm Yöntemlerinin Açıklanması

Uygulama şu an geliştirilme aşamasında olup şu an Android Emulator üzerinde çalışmaktadır. Şu an alınan verilerde Image Stitching işlemi başarılı bir şekilde gerçekleştirilebilmektedir. Ancak şu an alınan sonuçlar daha da iyileştirilmiş bir hale getirilebilir. Bunun için uygulamaya Önerilen yöntemin geliştirilmesi/yeniden kullanımı başlığında belirtiler iyileştirmeler denenecektir ve sonuçlar izlenip yeni iyileştirmelere gidilecektir.



## 1.2. Yazılım Çözümü



Görüntü Birleştirme probleminin çözümü için önerilen yazılım çözümü, OpenCV kütüphanesi üzerine inşa edilmiş bir Qt&QML uygulamasını içermektedir. Bu çözüm, görüntüleri özelliklerine göre eşleştiren, homografi tahmini yapan ve son olarak görüntüleri birleştiren bir dizi adımdan oluşmaktadır. Çözüm, performans açısından optimize edilmiş algoritmalar kullanır. Ayrıca, görüntü birleştirme sürecinde oluşabilecek hataları yönetmek

için detaylı bir hata işleme mekanizması içerir. Özellikle, Qt'nin döndürdüğü hata kodlarına dayalı olarak, kullanıcıya anlamlı hata mesajları sunar. Asenkron işlemleri düzenlemek ve kullanıcıya etkileşimli bir deneyim sunmak amacıyla Qt de bulunan Signal Slot yapısı kullanılmıştır. Bu sayede, görüntü birleştirme işlemleri arka planda çalıştırılırken uygulamanın tepki vermesi sağlanır.

### 1.2.1. Önerilen Yöntemin Test ve Kullanım Süreçlerine Ait Kullanım UseCase-Çizeneği

Uygulamadaki kullanıcının ana hedefinde kullanıcının, farklı bakış açılarından çekilmiş birden çok görüntüyü birleştirerek panoramik bir görüntü elde etmesi amaçlamaktadır.

Kullanıcı, görüntü birleştirme işleminde istediği sonuca ulaşır. Görüntüler doğru bir şekilde birleştirilir ve panoramik bir görüntü elde edilir ve uygulama, kullanıcıya başarılı bir birleştirme gerçekleştirildiği konusunda bilgi verirse uygulama başarılı bir senaryo geçirmiştir demektir.

Oluşabilecek hata senaryolarında ise:

**Yetersiz Görüntü Sayısı:** Kullanıcı, birleştirmek için yetersiz sayıda görüntü eklerse, uygulama bu durumu kullanıcıya bildirir.

**Homografi Tahmini Başarısızlık:** Homografi tahmini başarısız olursa, kullanıcıya hata mesajı gösterilir ve birleştirme işlemi durdurulur.

### Kullanım Süreci

#### • Hazırlık Aşaması

Kullanıcı, uygulamayı indirir ve kurar.

Görüntü birleştirme uygulamasını başlatır.

#### • Giriş Aşaması

Kullanıcı, birleştirmek istediği görüntüleri seçer.

Seçilen görüntüler uygulamaya yüklenir.

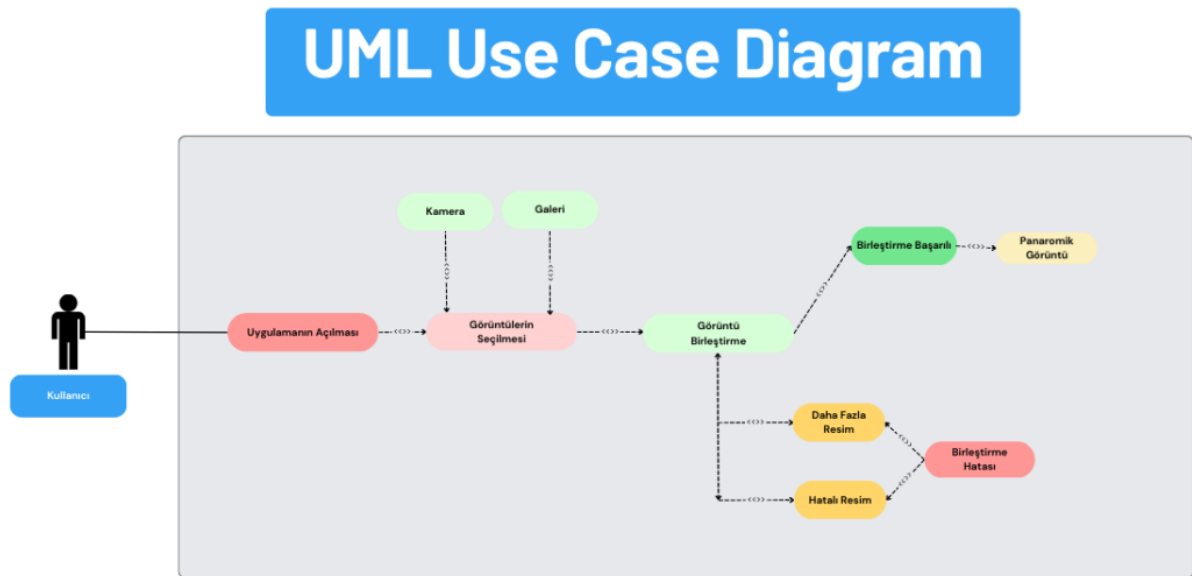
### • İşlem Aşaması

Özellik eşleştirme, homografi tahmini ve görüntü birleştirme işlemleri otomatik olarak başlar. Kullanıcı, bu süreçleri takip eder ve sonuçları gözlemleyebilir.

### • Sonuç Aşaması

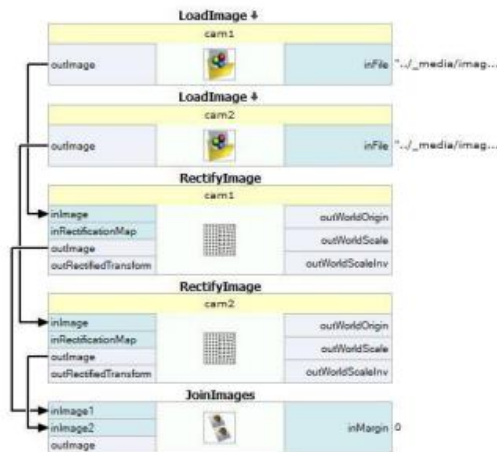
Birleştirilen görüntü kullanıcıya gösterilir.

Kullanıcı, sonucu değerlendirir ve gerekirse tekrar birleştirme işlemi yapabilir.



### 1.2.2. Kullanılan Teknolojileri İçeren Yazılım Bileşen-Component-Çizeneği

Bu bölümde, önerilen görüntü birleştirme yönteminin yazılımsal uygulamasında kullanılan teknolojileri içeren bir bileşen-bileşen çizeneği sunulacaktır.

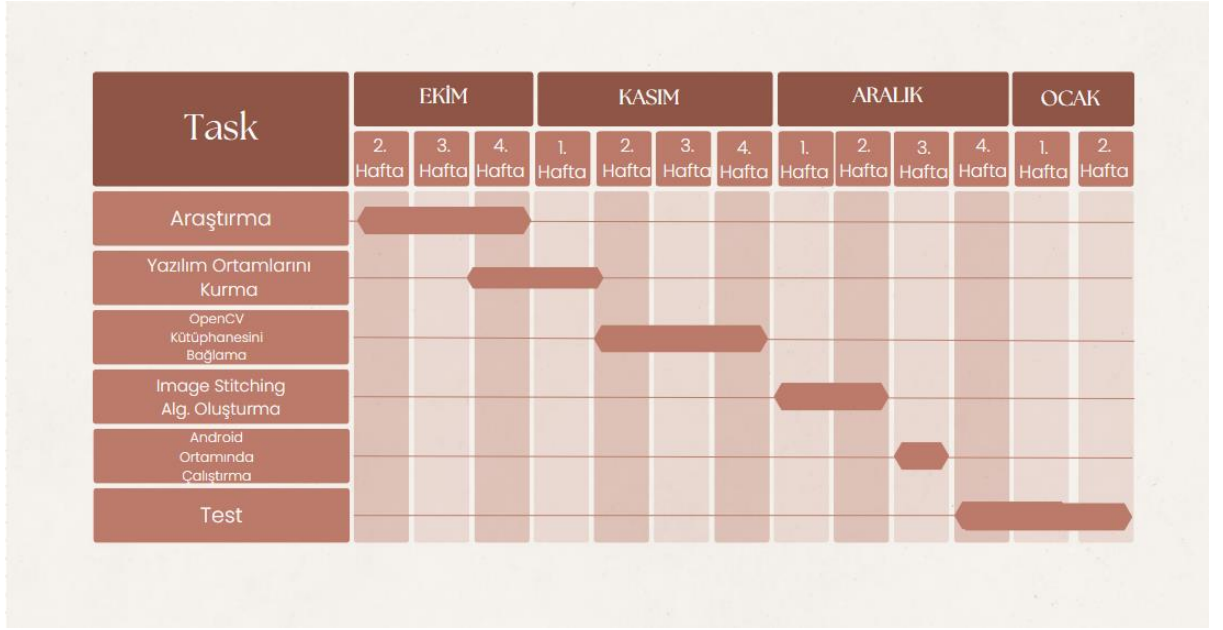


## 2. Çözüm İçin Geliştirilen Yöntemler ve Bunların Özgünlüğe Katkısı

Bu problemin ele alınmasındaki yaklaşımda hali hazırda yapılmış olan bir projenin tekrar yapılması gerçekleştirilmektedir. Göz retina fotoğraflarının birleştirilme işlemi için bu projede özgün olarak yapılması gereken işlemlerden birisi de maskeleme işlemleri olabilecektir. Burada veri setinden gelen düşük kaliteli fotoğrafların önce işlenerek yüksek kaliteli hale getirilmesi planlanmaktadır. Bu da daha önceki çözümlerden daha iyi sonuçlar verebilen özgün bir proje olabilmesini sağlayacaktır.

## 3. Proje Takvimi

Projenin GANT Diyagramı aşağıda verilmiştir. Bu diyagramda projenin kilometre taşları ve tarihleri belirtilmiştir.



## 4. Referanslar

- [1] J. Zhang, G. Chen, and Z. Jia, "An image stitching algorithm based on histogram matching and SIFT algorithm," Int. J. Pattern Recognit. Artif. Intell., vol. 31, no. 4, pp. 1–14, 2017, DOI: 10.1142/S0218001417540064.
- [2] M. Z. Bonny and M. S. Uddin, "Feature-based image stitching algorithms," IWCI 2016 - 2016 Int. Work. Comput. Intell., no. December, pp. 198–203, 201, DOI: 10.1109/IWCI.2016.7860365.
- [3] S. Mistry and A. Patel, "Image Stitching using Harris Feature Detection," Int. Res. J. Eng. Technol., vol. 03, no. 04, pp. 1363–1369, 2016, [Online]. Available: www.irjet.net.
- [4] R. Szeliski, "Image alignment and stitching: A tutorial," Found. Trends Comput. Graph. Vis., vol. 2, no. 1, pp. 1–10, 2006, DOI: 10.1561/0600000009.
- [5] K. Joshi, "OPEN ACCESS A Survey on Real-Time Image Stitching," vol. 10, no. 5, pp. 19–24, 2020, DOI: 10.9790/9622-1005011924.



[6] W. LYU, Z. ZHOU, L. CHEN, and Y. ZHOU, “A survey on image and video stitching,” *Virtual Real. Intell. Hardw.*, vol. 1, no. 1, pp. 55–83, 2019, DOI: 10.3724/sp.j.2096-5796.2018.0008.

[7]OpenCV. (2023). OpenCV Documentation. Retrieved from <https://docs.opencv.org/4.x/>