



**T.C.**  
**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Bilgisayar Mühendisliği Tasarım Çalışması II**

**Zeynep KOTAN**

**PROJE II ÇALIŞMASI**

**PROJE II DANIŞMANI : Öğr. Gör. Yusuf MUŞTU**

**BİLECİK**

**13 Ocak 2019**



**T.C.**  
**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**Bilgisayar Mühendisliği Tasarım Çalışması II**

**Zeynep KOTAN**

**PROJE II ÇALIŞMASI**

**PROJE II DANIŞMANI : Öğr. Gör. Yusuf MUŞTU**

**BİLECİK**

**13 Ocak 2019**

## **ÖZET**

### **Projenin Amacı**

Bu proje sayesinde mühendislik ve teknik eğitimde sıkça kullanılan MATLAB uygulamasının Web tabanlı olarak gerçekleştirilmesi hedeflenmiştir. Web sayfasında File Upload işlemi yaparak seçtiğimiz resim Matlab'a gönderilir. Görüntü işleme ile resimdeki kişi sayısı bulunur. Ardından kişi sayısı Web'de yazdırılır.

### **Projenin Kapsamı**

Matlab uygulaması olarak günümüzde yaygın olan görüntü işleme esas alınmıştır. Görüntü İşlemenin çok çeşitli ve farklı uygulamaları olmakla birlikte görüntü işleme, pek çok alanda kullanılmaktadır. Söz konusu yoğun kullanım alanlarının internet üzerinde icra edilmesi ile görüntü işleme kapsamında web tabanlı çeşitli proje, çalışma ve araştırma gibi uygulamalar gerçekleştirilebilir. Ayrıca arayüzün web tabanlı tasarlanmasıyla, yöntemin yaygınlaştırılması, zaman kaybını ortadan kaldırarak birden fazla kullanıcının eş zamanlı kullanabilmesi amaçlanmıştır.

## **ABSTRACT**

### **Project Objective**

Thanks to this project has been targeted frequently used in education of engineering and technical realization as a Web-based MATLAB applications. File Web page choose Upload making process images sent to MATLAB. The number of people located in the picture with the image processing. Then the number of people is printed on the web[5].

### **Scope of Project**

Today as Matlab image processing applications have also been widespread basis. While a variety of different image processing and image processing applications, however, are used in many fields. Said intensive use of the area covered by the image processing to be carried out on the internet with various web-based projects, such as studies and research carried out applications. In addition, by designing the web-based interface, dissemination methods, eliminating the time-consuming aimed multiple users can use simultaneously[5].

## **TEŞEKKÜR**

Bu projenin başından sonuna kadar hazırlanmasında emeği bulunan ve beni bu konuya yönlendiren saygıdeğer hocam ve danışmanım Sayın Öğr. Gör. Yusuf MUŞTU'ya tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

**Zeynep KOTAN**

13 Ocak 2019

# İÇİNDEKİLER

<b>ÖZET</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>TEŞEKKÜR</b>	<b>iv</b>
<b>ŞEKİL LİSTESİ</b>	<b>vii</b>
<b>1 GİRİŞ</b>	<b>1</b>
<b>2 MATERYAL VE METODLAR</b>	<b>3</b>
2.1 MATLAB . . . . .	3
2.2 VISUAL STUDIO . . . . .	4
2.3 POSTMAN . . . . .	4
<b>3 ASP.NET WEB API NEDİR?</b>	<b>6</b>
<b>4 GÖRÜNTÜ İŞLEME NEDİR?</b>	<b>8</b>
4.1 ÇALIŞMA MANTIĞI NEDİR? . . . . .	8
4.2 GÖRÜNTÜ İŞLEMENİN AMACI NEDİR? . . . . .	9
4.3 GÖRÜNTÜ İŞLEME TÜRLERİ . . . . .	9
4.4 GÖRÜNTÜ İŞLEME TEKNOLOJİSİNİN KULLANILDIĞI ALANLAR	10
<b>5 WEB API İLE GÖRÜNTÜ İŞLEME UYGULAMASI</b>	<b>11</b>
5.1 YÜZ TANIMA NEDİR? . . . . .	11
5.2 MATLAB'DA DLL OLUŞTURMA . . . . .	12
5.3 VISUAL STUDIO'DA FILE UPLOAD . . . . .	14
5.4 VISUAL STUDIO'DA MATLAB FONKSİYONLARINI ÇALIŞTIRMA	15
5.5 POSTMAN'DE WEB API TESTİ . . . . .	16
<b>6 SONUÇLAR VE ÖNERİLER</b>	<b>17</b>
<b>7 EKLER</b>	<b>18</b>
7.1 MATLAB KODLARI . . . . .	18

7.2	VISUAL STUDIO KODLARI . . . . .	19
7.2.1	HomeController.cs . . . . .	19
7.2.2	Index.cshtml . . . . .	21
7.2.3	DefaultController.cs . . . . .	23
<b>KAYNAKLAR</b>		<b>26</b>
<b>ÖZGEÇMİŞ</b>		<b>27</b>

## ŞEKİL LİSTESİ

1	Postman Arayüz Ekranı . . . . .	5
2	Postman Veri Gönderme . . . . .	5
3	Postman Döner Veriler . . . . .	6
4	Yüz Bulma Kodu . . . . .	11
5	Bulunan Yüzleri Klasöre Atma ve Kişi Sayısını Bulma Kodu . . . . .	12
6	Compiler Penceresi . . . . .	13
7	Dll Oluşturma . . . . .	13
8	Index Sayfası Kodları . . . . .	14
9	Index Sayfası . . . . .	14
10	Dll'den Resimdeki Kişi Sayısını Çekme . . . . .	15
11	Index Sayfasının Son Hali . . . . .	15
12	Web Api Kodları . . . . .	16
13	Postman'de Test . . . . .	16



# 1 GİRİŞ

Günümüzde bilgisayarlar yaşantımızın vazgeçilmez bir parçası haline gelmiştir. Akla gelebilecek her alanda bilgisayarın kullanımı görülmektedir. Özellikle bu çalışmada yapılan uygulama göz önüne alındığında bilgisayar destekli eğitim, araştırma, görüntü işleme ve benzetim uygulamaları açısından bilgisayarların kullanıcılara çok büyük oranda kolaylık sağladığı açık biçimde görülmektedir. Bilgisayar kullanımının getirdiği kolaylıklar yanında daha birçok avantajları bulunmaktadır. Bu avantajlardan bazıları bilgisayar destekli analiz ve benzetim arayüzleri temelinde sıralanırsa;

- Denenmesi riskli ve tehlikeli olan olayların deneysel çalışmaya ihtiyaç duyulmadan incelenmesi sağlanır,
- İşlemlerin otomatik olarak bilgisayar ortamında gerçekleştiriliyor olması zaman ve maliyetten tasarruf edilmesini ve en az hata ile daha verimli bir çalışma yapılmasını sağlar,
- Kullanıcı sayısında sınırlama olmaksızın işlemlerin defalarca tekrarlanmasına olanak sağlar,
- Elde edilen sayısal veriler kolaylıkla grafik haline dönüştürülebilir,
- Yapılan işlemler hakkında görsel bir değerlendirme ve karşılaştırma yapılabilir.

Sıralanan tüm bu avantajlar sayesinde bilgisayarlar ile gerçekleştirilen benzetim ve analiz uygulamaları bilgisayar destekli eğitim, uzaktan eğitim, araştırma-geliştirme alanlarında oldukça yaygın biçimde kullanılmaktadır. Bu amaçla yapılmış arayüz temelli çalışmalarda birkaç farklı platformdan yararlanılabilir. Örneğin MATLAB GUI ile hazırlanan bu arayüzler MATLAB'ın gelişmiş analiz ve grafik özelliklerini kullandıklarında çok esnek ve kullanışlı bir yapıya sahiptirler. Ancak bu platformda hazırlanan arayüzlerin çalışması için o bilgisayarda MATLAB programının yüklü olması gerekmektedir. MATLAB'a olan bu bağımlılık hem maliyet hem de hazırlanan arayüzün taşınabilirliği açısından dezavantaj oluşturmaktadır. MATLAB GUI için ortaya konan dezavantajların

giderilmesi için daha genel kullanıma yönelik platformlar (Flash, .NET gibi) ile arayüzler hazırlanabilir. Bu şekilde yapılan bir uygulama herhangi bir bilgisayarda farklı bir yazılıma ihtiyaç duymaksızın çalıştırılabilir bir yapıya sahip olduğundan hem maliyet hem de taşınabilirlik açısından avantaj sağlar. Ayrıca .NET platformu C#, ASP.NET, VB.NET gibi diller için ortak bir platform olduğundan uygulamanın geliştirilmesi sırasında bu dillerden birlikte yararlanılabilme imkanı vardır.

Arayüz tasarımında kullanılabilecek üçüncü yaklaşım ise MATLAB ve .NET platformlarının avantajlarına sahip olan bir yapıdır. Hatta bu yaklaşım ile bu avantajları bir adım daha öteye taşıyarak web temelli yapıya kavuşturmak ve yapılan uygulamanın yaygınlaşmasını sağlamak da mümkündür.

Son yıllarda özellikle mühendislik eğitime yönelik internet tabanlı eğitimsel arayüz çalışmaları hız kazanmıştır. Konuyla ilgili literatürde çok sayıda çalışma mevcuttur. Bu çalışmalar incelendiğinde MATLAB programının gerçekleştirilen uygulamalarda temel araçlardan bir tanesi olduğu görülmektedir. MATLAB programının özelliklerini web temelli uygulamalara aktarmak için günümüzde MATLAB Builder NE ve MATLAB Web Figure araçları kullanılmaktadır. Bu araçlar ile MATLAB'ta hazırlanan fonksiyonlar .NET bileşenlerine dönüştürülerek kullanılmaktadır. Bu yaklaşıma örnek olarak hazırlanmış olduğum kişi sayısı analizini verilebilir.

## 2 MATERYAL VE METODLAR

Bu bölümde kullanılan materyal ve metodlardan bahsedilecektir. Projede kullanılan materyaller:

- MATLAB
- VISUAL STUDIO
- POSTMAN

Seçilen materyallerin neden seçildiği hakkında kısaca bilgilendireyim. Matlab Görüntü İşleme alanında uygulama geliştirmek için gereklidir. Visual Studio ise File upload işlemini yaparak Matlab da oluşturduğumuz dll'e resmi göndererek sonucu yazdırma işlemini yapıyor. Postman ise Visual Studio'da yazılan Api'yi test etmek, paylaşmak, doküman ve monitör elde etmek için kullanılan arayüzdür.

### 2.1 MATLAB

MATLAB, genellikle pozitif bilim ve mühendislik hesaplamaları için kullanılan bir bilgisayar programıdır. Amerika Birleşik Devletleri merkezli MathWorks firması tarafından geliştirilen MATLAB, aynı zamanda bir programlama dilidir. İngilizce "Matrix Laboratory" kelimelerinin birleştirilmesi ile oluşmuş olan MATLAB, isminden de anlaşılacağı gibi matris tabanlı bir çalışma sistemine sahiptir. Lineer cebir, istatistik, optimizasyon, nümerik analiz, optimizasyon, fourier analizi gibi pek çok matematiksel hesaplamaların etkili ve hızlı şekilde yapılmasına olanak sağlayan MATLAB programı aynı zamanda 2D ve 3D grafik çizimi için de kullanılır.

MATLAB ile kullanıcılar kendi programlarını hazırlayabilirler. Matrisler ve onların etkileşim içinde olduğu fonksiyonlarla programlama yapılmasına izin veren MATLAB ile çok karmaşık matematik hesaplamaları bile birkaç saniye içinde tamamlanır. Temel programlama fonksiyonları ile benzer fonksiyonların kullanılabildiği MATLAB ile etkili ve pratik programlar hazırlanabilir.

C, Java gibi programlama dillerindeki dizilerin kullanımı ile aynı mantıkla matrislerin kullanıldığı MATLAB programında bir, iki veya daha fazla boyutta matrisler ile çalışmak mümkündür. MATLAB ile temel matematik fonksiyonlarının iki ve üç boyutlu grafikleri çizilebilir. Polinomlar, parboller, sinüs dalgaları başta olmak üzere her tür iki ve üç boyutlu matematiksel grafik MATLAB ile elde edilebilir[1].

## 2.2 VISUAL STUDIO

Microsoft Visual Studio, Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework ve Microsoft Silverlight tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve Windows Forms uygulamaları, web siteleri, web uygulamaları ve web servisleri ile birlikte konsol ve grafiksel kullanıcı arayüzü uygulamaları geliştirmek için kullanılır.

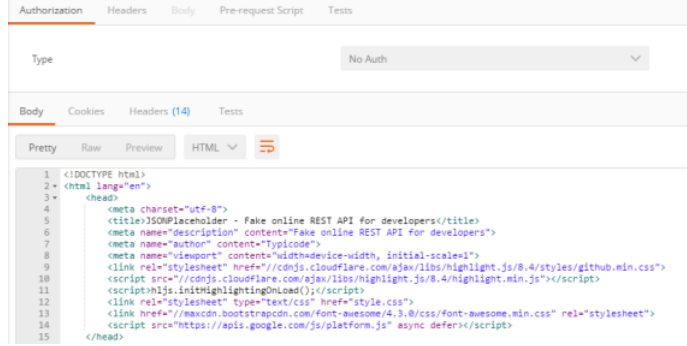
Visual Studio IntelliSense'in yanı sıra "code refactoring" destekleyen bir kod editörü içerir. Entegre hata ayıklayıcı, hem kaynak-seviyesinde hem de makine-seviyesinde çalışır. Diğer yerleşik araçlar, GUI uygulamaları, web tasarımcısı, sınıf tasarımcısı ve veritabanı şema tasarımcısı yaratabilmek için bir form tasarımcısı içerir. Hemen hemen her işlevsellik düzeyinde dahil olmak üzere, kaynak kontrol sistemleri için destek (Subversion ve Visual SourceSafe gibi) sunan eklentileri kabul eder.

Visual Studio, değişik programlama dillerini destekler, bu da kod editörü ve hata ayıklayıcısının neredeyse tüm programlama dillerini desteklemesini sağlamaktadır. Dahili diller C/C++ (Görsel yoluyla C++), VB.NET (Visual Basic .NET üzerinden), C# (Visual C# ile), ve F# (Visual Studio 2010 itibarıyla) içermektedir[2].

## 2.3 POSTMAN

Postman, API testleri için kullanılan Back-End yazılım geliştirenler için üretilmiş Rest client uygulamasıdır. Bir API var diyelim bu API'nin yeteneklerini ve çalışma hızını

görmek , durumunu test etmek , veri içerisinde gelen değerleri sorugulamak için postman-  
dan yararlanılır.



1. Authorization : Kimlik doğrulama işlemi için buradan ona uygun formatta değerler girilir.

2. Header : Request bilgileriyle birlikte gönderilecek header değerleri girilir.

3. Body : Aktif hale gelmesi için menüden "POST" seçilmesi gerekmektedir. Ve post değerleri karşı tarafa.4 farklı şekilde gönderilebilir.

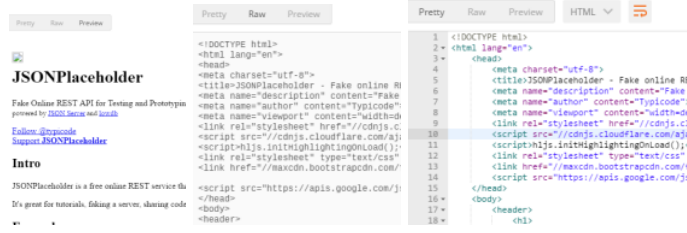
a) form-data b) x-www-form-urlencoded c) raw d) binary

Raw'ı seçip ve Json formatında veriler set edilir.



4. Response Body : Body tabında dönen değerler geliyor ve çıktığı 3 farklı formatta veriyor.

a) Pretty b)Raw c)Preview



Şekil 3: Postman Dönen Veriler

### 3 ASP.NET WEB API NEDİR?

Web Api'ye geçmeden önce Api nedir ondan bahsedeyim. Api açılımı "Application Programming Interface" olan Türkçe'de uygulama geliştirme arayüzü anlamına gelir ve sahip olduğumuz service veya verileri dış dünyaya açıp başka uygulamaların-platformların kullanımına sunmak için belli kurallar çerçevesinde tanımlamalar yaptığımız arayüz'dür.

Asp .Net Web Api ise farklı türde sayısız client (browsers, mobile phones, tablets, pc, etc.) tarafından consume edilebilen HTTP protokolü üzerinden haberleşebilen servisler oluşturmak için kullanılan bir framework şeklinde tanımlayabiliriz. Asp .net MVC ile routing, controllers, action results, filter, model binders gibi ortak feature'lara sahip olduklarından bir takım benzerlikler göstermektedir ancak MVC Framework'ün bir parçası değildir. Asp .net Web Api Core Asp .Net'in bir parçasıdır ve MVC veya diğer web application türleri ile birlikte kullanılabilir. Aynı zamanda bütün bunlardan bağımsız standalone Web services application olarakta kullanılabilir.

Günümüz dünyasında teknolojinin gelişmesiyle birlikte firmalar artık web tabanlı uygulamalar üzerinden müşterilerine tam olarak ulaşamaz hale geldiler. İnsanlar artık günlük hayatlarının nerdeyse %50 sini akıllı telefonlar, tablet pc vs ile geçiriyorlar ve bu cihazlarda insanların hayatlarını kolaylaştıracak olan milyonlarca uygulama mevcut. Bunların yanında birde IOT ile birlikte gelecek 5 yılda dünyada 30 milyara yakın internete bağlanabilen cihazlar olacağından bahsediliyor ve buda belki milyonlarca Api geliştirmesi demek.

Firmalar veya uygulama geliştiriciler müşterilere daha kolay ve hızlı bir şekilde ulaş-

mada kullanmak için servislerini ve sahip oldukları verilerin bir kısmını browserlar ya da internete bağlanabilen bu akıllı cihazlar tarafından consume edilebilmeleri için Api'lar geliştirmeleri gerekmektedir. Çünkü Api'lar yapısı gereği bütün programlama dilleri tarafından ortak kabul görmüş medya tiplerini (XML-JOSN..etc.) response olarak alıp gerekli parse işlemlerinden sonra kolayca kullanabilir.

Web Api sahip olduğunuz veri ve servisleri birçok farklı cihazda kullanıma sunmak için expose edebilmenizi sağlayan şahane bir framework ve dahası Web Api .Net Framework üzerinde RESTful servisler inşa etmenizi sağlayacak ideal bir open source platform. WCF Rest service'lerinin aksine Web Api HTTP protokolünün bütün özelliklerini kullanır (URIs, request/response headers, caching, versioning, çeşitli content format'ları) WCF Rest Service'lerinde yapıldığı gibi farklı cihazlar için extra config ayarları vs yapmamıza da gerek bulunmamaktadır. Request'i yapılırken dönmesi gereken response'un XML mi yoksa JSON formatında mı olacağına client'ın seçimine bırakılmıştır. Çünkü Web Api birden fazla medya formatında response dönebilmektedir.

#### Başlıca Web API Özellikleri

- Http Get, Post, Put ve Delete metodlarıyla çalışabildiğinden CRUD işlemlerini destekler,
- Response'larda HttpStatusCode ve Accept Header parametreleri bulunur,
- Response'lar kullanıcının istediği türde MediaTypeFormatter tarafından formatlanabilir,
- OData desteği bulunmaktadır ve Query yazması oldukça kolaydır,
- Bir uygulama içerisinde veya IIS üzerinde host edilebilir,
- MVC'nin bazı özelliklerini taşır (routing, controllers, action results, filter, model binders)

Neden Web Api'ı Seçmeliyiz ?

- Bir web service'e ihtiyacınız varsa ve soap'a ihtiyacınız yoksa en iyi seçenek Web Api dir,
- Geliştirme sürece WCF de olduğu kadar zahmetli ve sıkıntılı değildir,
- Http tabanlı olduğundan Rest-ful servisler geliştirmek için en iyi seçenektir,
- Exception ve Cache mimarileri oldukça performanslı ve yönetilebilir dir,
- Open Source olduğundan sürekli olarak geliştirilip yeni feature'lar eklenmektedir,
- Microsoft yetkilileri Web Api sunumlarından birinde şuna benzer bir şey söyledi "Biz daha iyisini yapana kadar en iyisi bu..!" bu demek oluyor ki tam destek[3].

## 4 GÖRÜNTÜ İŞLEME NEDİR?

Görüntü işleme, bir görüntüyü dijital form haline getirmek ve bazı işlemleri gerçekleştirmek için geliştirilmiş bir görüntü elde etmek veya ondan bazı yararlı bilgiler çıkarmak için kullanılan bir yöntemdir. Bu, video çerçevesi veya fotoğraf gibi bir girdinin görüntünün olduğu ve çıktı ile ilişkili görüntü veya karakteristik olabilen bir sinyal tutma türüdür. Genellikle Görüntü İşleme sistemi, görüntüleri önceden tanımlanmış sinyal işleme yöntemleri uygularken iki boyutlu sinyaller olarak işler.

### 4.1 ÇALIŞMA MANTIĞI NEDİR?

Görüntü işleme temel olarak aşağıdaki üç adımı içerir.

1. Görüntünün optik tarayıcı veya dijital fotoğraflarla alınması.
2. Veri sıkıştırma, görüntü iyileştirmeye ve uydu fotoğrafları gibi insan gözleri için olmayan lekelenme kalıplarını içeren görüntüyü analiz etme ve kullanma.



3. Çıktı, sonuçların imge analizine dayanan görüntü veya rapor değiştirilebilen son aşamasıdır.

## 4.2 GÖRÜNTÜ İŞLEMENİN AMACI NEDİR?

Görüntü işlemenin amacını beş maddede özetleyebiliriz:

1. Görselleştirme: görünmeyen nesneleri gözlemlemek için kullanırız.
2. Görüntü Keskinleştirme Ve Restorasyon: Görüntü netleştirme ve daha iyi görüntü almak için kullanırız.
3. Görüntü Alımı: İlgi çeken görüntüler için kullanırız.
4. Desen Ölçümü: Görüntüdeki çeşitli nesneleri tespit etmek için kullanırız.
5. Resim Tanıma: Görüntüdeki çeşitli nesneleri ayırt etmek için kullanırız.

## 4.3 GÖRÜNTÜ İŞLEME TÜRLERİ

Görüntü işleme için kullanılan iki yöntem Analog Görüntü İşleme ve Dijital Görüntü İşleme yöntemidir. Baskı ve fotoğraf gibi basılı kopyalar için analog veya görsel görüntü işleme teknikleri kullanılabilir. Görüntü analistleri, bu görsel teknikleri kullanırken yorumlamanın çeşitli temellerini kullanırlar. Görüntü işleme sadece incelenmesi gereken alanla sınırlı değil analist bilgisi üzerine de yapılabilir. Görsel tekniklerle görüntü işleme alanındaki bir diğer önemli araç da birlikteliktir. Bu yüzden analistler, kişisel bilgi ve teminat verilerini bir arada görüntü işleme işlemine tabi tutarlar[4].

Dijital İşleme teknikleri dijital görüntülerin bilgisayarlarla düzenlenmesine yardımcı olur. Uydu platformundan görüntü algılayıcılarına ait ham veriler eksiklik içerdiğinden bu kusurları aşmak ve bilginin özgünlüğünü elde etmek için, çeşitli işlem ve aşamalardan geçmek zorundadır. Her türlü verinin dijital tekniği kullanırken geçmesi gereken üç genel aşama;

1. Ön İşleme
2. Geliştirme ve Görüntüleme
3. Bilgi Çıkarımı'dır.

#### **4.4 GÖRÜNTÜ İŞLEME TEKNOLOJİSİNİN KULLANILDIĞI ALANLAR**

- Ürün Sayma ve Hata Tespiti
- Uydu Görüntüleri Üzerinde Nüfus Yoğunluğu, Çevre Kirliliği Ve Benzeri Çevresel Şartların Tespiti
- Kalite, Doğruluk ve Yüzey Analizi
- Askeri Endüstri (Denizaltı Sonic Dalga Taramaları), Sualtı Görüntüleme
- Robot Kol Yönetimi
- Eksik, Hatalı Parça veya Üretim Kontrolü
- Robotik, Trafik, Astronomi, Radar, Gazete Ve Fotoğraf Endüstrisi Uygulamaları
- Fizik, Sanat, Biyomedikal Alanlar
- Uzaktan Algılama Uygulamaları
- Uydu Görüntüleri Üzerinde Hava Gözlem ve Tahmin Uygulamaları

## 5 WEB API İLE GÖRÜNTÜ İŞLEME UYGULAMASI

### 5.1 YÜZ TANIMA NEDİR?

Bir yüz dağarcığı içerisinde bir kişinin yüzünü tespit edip o kişiyi bulma işlemine yüz tanıma denir. Yüz tanıma işlemi öncelikle yüzün tespiti ve daha sonra tespit edilen bu yüzün veritabanındaki yüz dağarcığıyla çeşitli yöntemleri (PCA, LDA, EP, ICA vb.) kullanarak bulma işlemidir.

Matlab’da kişi sayısını bulabilmek için öncelikle resimdeki yüzleri bulmak gerekmektedir. Bir resim içinde yüzün konumunu tespit etmek için Matlab’ın içinde bulunan Viola-Jones algoritması kullanarak nesneleri algılama işlemi yapan Cascade nesne dedektörünü kullanılır. Cascade nesne dedektörü uygulanan resim içerisinde bulunduğu yüzlerin resim üzerindeki x ve y eksenine göre koordinatlarını vermektedir. Temel olarak dedektör yüz algılamak üzere yapılandırılmış, fakat diğer nesne türleri için yapılandırılabilir.

```
function output = cucu(a)

faceDetector=vision.CascadeObjectDetector('FrontalFaceCART'); %yüz bulma dedektörü oluşturuluyor

img=imread(a);

BB=step(faceDetector,img); % resimlerin x y eksenine göre kordinatları BB değişkenine atılıyor

iimg = insertObjectAnnotation(img, 'rectangle', BB, 'Face'); %kordinatlardaki yüzler çerçeveye alınıyor
```

Şekil 4: Yüz Bulma Kodu

Bulunan yüzleri çerçeveye aldıktan sonra klasöre kaydedilir. Burada klasöre yüzleri yerleştirmek için isim verilir. Verilen ismi counter adında bir sayaç oluşturarak sayısal olarak tanımlandı. For döngüsü içinde tekrardan ksayisi adında sayaç oluşturarak kişi sayısı bulunur.

```
N=size(BB,1); %BB değişkeninin satır sayısını N değişkinine ata
handles.N=N;
counter=1;
ksayisi=0;

for i=1:N
    face=imcrop(img,BB(i,:));

    baseDir = 'C:\P\TestDatabase\'; %kesilen yüzlerin atılacağı klasör

    newName = [baseDir num2str(counter) '.png'];
    handles.face=face;

    while exist(newName,'file')

        counter = counter + 1; %sayaç belirleyerek kesilen yüzlere integer değer vererek kaydedilir
        newName = [baseDir num2str(counter) '.png'];

    end

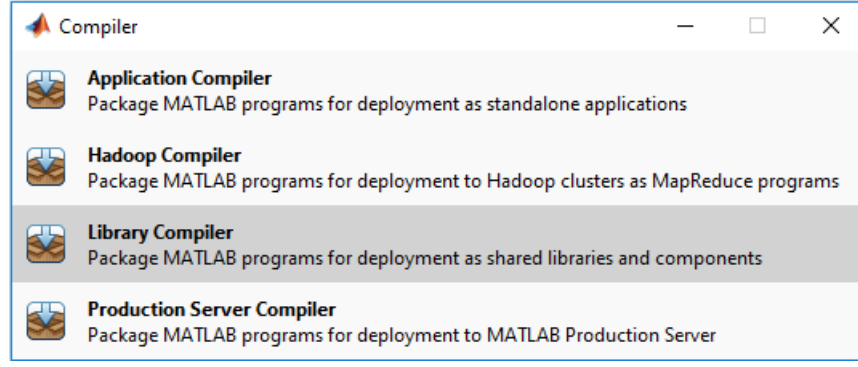
    ksayisi=ksayisi+1;
    fac=imresize(face,[200,180]);
    imwrite(fac,newName);
    output=ksayisi;

end
```

Şekil 5: Bulunan Yüzleri Klasöre Atma ve Kişi Sayısını Bulma Kodu

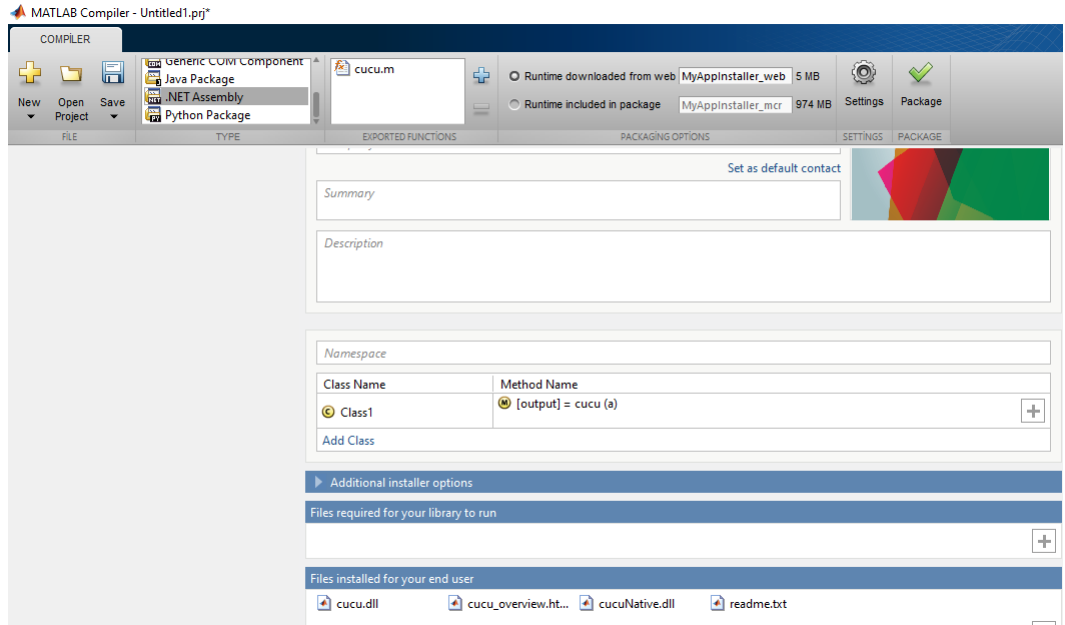
## 5.2 MATLAB'DA DLL OLUŞTURMA

Matlab'da dll oluşturmak için Command Window penceresine 'deploytool' yazılır. Daha sonra açılan Compiler penceresinde Library Compiler seçilir.



Şekil 6: Compiler Penceresi

Dll'i Visual Studio'da çalıştırmak için .Net Assembly'i seçilir. Daha sonra .m uzantılı Matlab dosyası eklenir. Package'ye tıklayarak dll'in oluşturulacağı klasör seçilerek işlem tamamlanmış olur.



Şekil 7: Dll Oluşturma

### 5.3 VISUAL STUDIO'DA FILE UPLOAD

Bu uygulamayı gerçekleştirmek için ilk olarak dosyadan resim belirlemek gerekir. Öncelikle projede HomeController içine Index adında bir action oluşturulur. Index'e sağ tıklayıp View oluşturularak projenin dosya yükleme işlemi gerçekleştirilir. Bu işlem için metodunu post, enctype'ı multipart/form-data olarak tanımlanır.

Div oluşturduktan sonra içerisine bir adet input > file ve bir adet button > submit kontrolleri yerleştirilir. Input > file kontrolünün name özelliği Image olarak ayarlandı. Bu tanım controller içerisinde kullanılır.

```
using (Html.BeginForm("Index", "Home", FormMethod.Post, new { enctype = "multipart/form-data" }))
{
    <div class="arkaresim">
        <div class="row" style="margin-top:50px;">
            <div class="col-md-6">
                <div class="form-group">
                    <label for="Image">Resim Seçiniz:</label>
                    <input type="file" class="form-control" name="Image" id="fileUpload" />
                </div>
                <div class="form-group">
                    <button type="submit" class="btn btn-primary "><i class="glyphicon glyphicon-upload"></i> Yükle</button>
                </div>
                <label>Kişi Sayısı:</label>
                <input type="text" class="form-control" disabled value="@ViewData["kisisayisi"]" />
            </div>
        </div>
    </div>
}
```

Şekil 8: Index Sayfası Kodları

Web Tabanlı Görüntü İşleme

Resim Seçiniz:

Dosya seçilmedi

Kişi Sayısı:

Şekil 9: Index Sayfası

## 5.4 VISUAL STUDIO'DA MATLAB FONKSİYONLARINI ÇALIŞTIRMA

HomeController'da yazılan dll'li çağırılır. Resmi belirledikten sonra yükle butonuna tıklayarak oluşturulan dll'e gönderilir.

```
[HttpPost]
0 references
public ActionResult Index(HttpPostedFileBase Image)
{
    //var fileName = Path.GetFileName(Image.FileName);
    if (Image != null)
    {
        var dosyaYolu = Path.GetFileName(Image.FileName);
        var yuklemeYeri = Path.Combine(Server.MapPath("~/Image"), dosyaYolu);
        Image.SaveAs(yuklemeYeri);
    }
    Class1 obj = new Class1();

    MArray output = obj.cucu("C:\\P\\xx\\bulut\\bulut\\Image\\" + Image.FileName);
    var sayi = output.ToString();
    ViewData["kisisayisi"] = sayi;

    return View();
}
```

Şekil 10: Dll'den Resimdeki Kişi Sayısını Çekme

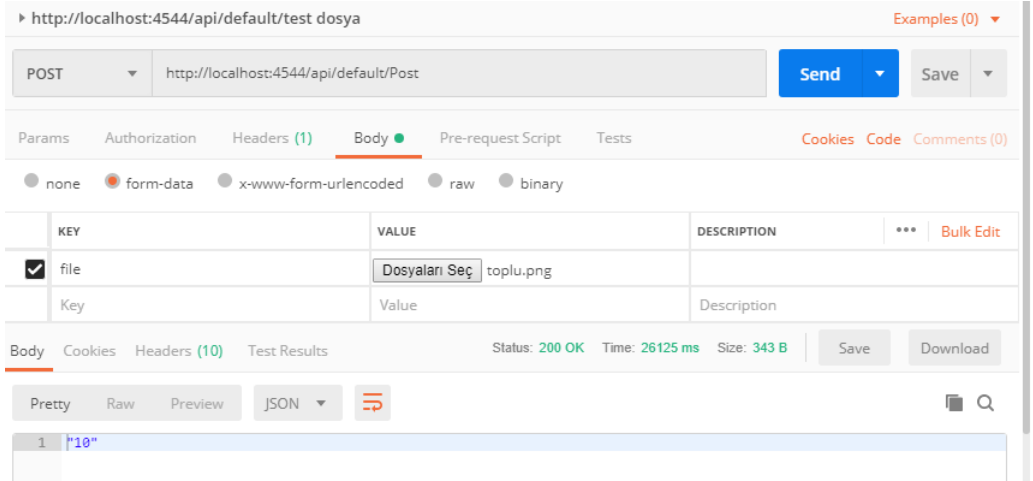
Şekil 11: Index Sayfasının Son Hali

## 5.5 POSTMAN'DE WEB API TESTİ

Web Api'de ilk yapılması gereken oluşturulacak Controller'ı Web Api2 formatında seçmektir. Daha sonra burada class oluşturularak sunucuda depolanacak resmin yolu saklanır. Matlab dll'ine gönderilecek resmin yolu yazılır ve sonucu döndürülür.

```
namespace Servis.Controllers
{
    0 references
    public class DefaultController : ApiController
    {
        1 reference
        public class talep
        {
            1 reference
            public string yol { get; set; }
        }
        [HttpPost]
        0 references
        public string test(talep t)
        {
            Class1 obj = new Class1();
            MimeType output = obj.cucu(t.yol);
            return output.ToString();
        }
    }
}
```

Şekil 12: Web Api Kodları



Şekil 13: Postman'de Test



## 6 SONUÇLAR VE ÖNERİLER

Hepimiz bilgisayar teknolojisinde ve görüntü işlemede hızlı gelişme ile ateşlenen devrimin ortasındayız. Bilinen inanca karşın, bilgisayarlar görüntü işleme ve analiz ile ilgili hesaplamalarda insanlarla eşleşemez. Ancak modern bilgisayarın artan gelişmişliği ve gücü ile hesaplama Von Neumann ardışık mimarinin ötesine geçecek ve optik yürütmeyi de düşünecektir. Paralel dağıtılmış bilgi işlem paradigmalarının, görüntü işleme sonuçları için tepkileri geliştirmesi beklenmektedir.

Ayrıyeten yeni bir teknoloji olan Bulut bilişim de gözleri üzerine çekmiştir.

Bulut bilişim, üçüncü parti yazılımlar tarafından desteklenen ortak bir hesaplama, depolama, ağ ve uygulama yazılımı platformudur. Bulut teknolojisi, kullanıcıların büyük miktarlardaki verileri işlemesi ve hesaplamaları yapabilmesi için soyutlanmış ve sanallaştırılmış bilişim kaynaklarını kullanmasını sağlar. Bulut bilişimin bir diğer önemli avantajı da, uzak mesafedeki araştırmacıların verileri paylaşmak yoluyla ortak çalışma yapabilmesidir. Bu çalışmada, bulut bilişim platformu üzerinde çalışan web tabanlı bir görüntü işleme uygulaması gerçekleştirilmek istenmiştir. Ancak karşılaşılan hatalar neticesinde projenin bulut bilişim kısmı gerçekleştirilememiştir.

## 7 EKLER

### 7.1 MATLAB KODLARI

Matlab’da bir resimdeki kişi sayısını bulabilmek için ilk olarak Casced nesne dedöktörü ile yüzler bulundu. Daha sonra bulunan yüzler TestDatabase adlı klasöre kydedilir. For döngüsü içerine sayaç konularak kaydedilen resim sayısı bulunur. Bu da resimdeki kişi sayısını verir.

```
function output = cucu(a)
faceDetector=vision.CascadeObjectDetector('FrontalFaceCART');
%yüz bulma dedektörü oluşturuluyor
img=imread(a);
BB=step(faceDetector,img);
%resimlerin x y eksenine göre
%kordinatları BB değişkenine atılıyor.
iimg = insertObjectAnnotation(img,'rectangle',BB,'Face');
%kordinatlardaki yüzler çerçeveye alınıyor
N=size(BB,1);
%BB değişkeninin satır sayısını
%N değişkinine atılır.
handles.N=N;
counter=1;
ksayisi=0;
for i=1:N
    face=imcrop(img,BB(i,:));
    baseDir = 'C:\P\TestDatabase\';
    %kesilen yüzlerin atılacağı klasör
    newName = [baseDir num2str(counter) '.png'];
    handles.face=face;
```

```

        while exist(newName,'file ')
            counter = counter + 1;
%sayaç belirleyerek kesilen yüzlere
%integer değer vererek kaydedilir.
            newName = [baseDir num2str(counter) '.png'];
        end
        ksayisi=ksayisi+1;
        fac=imresize(face,[200,180]);
%200-180 boyutundaki yüzler
%imwrite ile kaydedilir.
        imwrite(fac,newName);
        output=ksayisi;
    end
end

```

## 7.2 VISUAL STUDIO KODLARI

Visual Studio için ilk olarak dosyadan resim belirlemek gerekir. Öncelikle projede HomeController içine Index adında bir action oluşturulur. Index'e sağ tıklayıp View oluşturularak projenin dosya yükleme işlemi gerçekleştirilir. Bu işlem için metodunu post, enctype'ı multipart/form-data olarak tanımlanır.

Div oluşturduktan sonra içerisine bir adet input > file ve bir adet button > submit kontrolleri yerleştirilir. Input > file kontrolünün name özelliği Image olarak ayarlandı. Bu tanım controller içerisinde kullanılır.

### 7.2.1 HomeController.cs

```

using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Web;
using System.Web.Mvc;
using System.IO;
using cucu;
using MathWorks.MATLAB.NET.Arrays;
using System.Threading.Tasks;
using HttpPostedFileHelper;
namespace bulut.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
    }
}

[HttpPost]
public ActionResult Index(HttpPostedFileBase Imagee)
{
    if (Imagee != null)
    {
        var dosyaYolu=Path.GetFileName(Imagee.FileName);
        var yuklemeYeri=Path.Combine(Server.MapPath("~/Image"),dosyaYolu);
        Imagee.SaveAs(yuklemeYeri);
    }
    Class1 obj = new Class1();
    MWArray output=obj.cucu("C:\\P\\xx\\bulut\\bulut\\Image\\"
        +Imagee.FileName);
    var sayi = output.ToString();
    ViewData["kisisayisi"] = sayi;
    TempData["dosyayolu"] = Imagee.FileName.ToString();
}

```

```

        return View();
    }
}

```

### 7.2.2 Index.cshtml

```

@ {

    ViewBag.Title = "Index";

    string dosyayolu = TempData["dosyayolu"] as string;

} <style>

body {

    background-color:skyblue;

    background-repeat: no-repeat;

    background-attachment: fixed;

    font-family: Verdana, monospace, sans-serif;

    font-size: 12px;

    font-weight: bold;

    text-align: justify;

    min-height: 800px;

} div.arkaresim {

    width: 600px;

    height: 300px;

```

padding: 20px;

background-color: skyblue;

position: relative;

margin: auto;

}

</style>

@using (Html.BeginForm("Index", "Home", FormMethod.Post,

new { enctype = "multipart/form-data" }))

{ <div class="arkaresim">

<div class="row" style="margin-top:50px;">

<div class="col-md-6">

<div class="form-group">

<label for="Image">Resim Seçiniz:</label>

<input type="file" class="form-control" name="Imagee" id="fileUpload" />

</div>

<div class="form-group">

<button type="submit" class="btn btn-primary ">

<i class="glyphicon glyphicon-upload"></i>

Yükle</button>

</div>

```

<label>Kişi Sayısı:</label>

<input type="text" class="form-control"

disabled value="@ViewData["kisisayisi"]" />

</div> <div class="result">



</div>

</div>

</div>

}

```

### 7.2.3 DefaultController.cs

```

using cucu ;
using MathWorks.MATLAB.NET.Arrays ;
using System ;
using System.Collections.Generic ;
using System.Linq ;
using System.Net ;
using System.Net.Http ;
using System.Web ;
using System.Web.Http ;
namespace Servis.Controllers
{
    public class DefaultController : ApiController
    {
        public class talep
        {

```

```

        public string yol { get; set; }
    }
[HttpPost]
    public string test(talep t)
    {
        Class1 obj = new Class1();
        MWArray output = obj.cucu(t.yol);
        return output.ToString();
    }
    public string Post()
    {
        HttpResponseMessage result = null;
        var httpRequest = HttpContext.Current.Request;
        if (httpRequest.Files.Count > 0)
        {
            var docfiles = new List<string>();
            foreach (string file in httpRequest.Files)
            {
                var postedFile = httpRequest.Files[file];
                var filePath = HttpContext.Current.Server.MapPath("~/\"
                    + postedFile.FileName);
                postedFile.SaveAs(filePath);
                docfiles.Add(filePath);
            }
        }
        else
        {
            result = Request.CreateResponse(HttpStatusCode.BadRequest);
        }
        return "-1";
    }

```



}  
}

## KAYNAKLAR

- [1] CTAN,<http://www.wikizeroo.net/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTUFUTEFC>
- [2] CTAN,<http://www.wikizeroo.net/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTWljcm9zb2Z0X1Zpc3VhbF9T>
- [3] <http://www.canertosuner.com/post/Web-API-Nedir>
- [4] [https://www.mustafasaridal.com/goruntu-isleme/](https://www.mustafasaridal.com/goruntu-isleme/goruntu-isleme-teknolojisi-image-processing-nedir/)  
[goruntu-isleme-teknolojisi-image-processing-nedir/](https://www.mustafasaridal.com/goruntu-isleme/goruntu-isleme-teknolojisi-image-processing-nedir/)
- [5] <https://cevirsozluk.com/>

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Zeynep KOTAN  
**Uyruğu** : T.C.  
**Doğum Yeri ve Tarihi:** Aşkale /04.08.1997  
**Adres** : Güneşli 15 Temmuz mah. 1402.sokak no:19/2 Bağcılar/İST  
  
**Telefon** : 539 565 2406  
**e-mail** : zeynepkotan1@gmail.com

### EĞİTİM DURUMU

**Lisans Öğrenimi** : BŞEÜ Bilgisayar Mühendisliği Bölümü  
**Bitirme Yılı** :  
**Lise** : Ahi Evren Anadolu İHL

### İŞ DENEYİMLERİ

**Yıl** :  
**Kurum** :  
**Stajlar** : Staj1

### İLGİ ALANLARI:

**YABANCI DİLLER:** İngilizce:Orta Arapça:Başlangıç

**BELİRTMEK İSTEDİĞİNİZ DİĞER ÖZELLİKLER:**