**DAĞITIK SİSTEMLER İÇİN GÖRÜNTÜ İŞLEME WEB SERVİS UYGULAMASI**

**AN IMPLEMENTATION OF IMAGE PROCESSING WEB-SERVICES FOR DISTRIBUTED SYSTEMS**

Recep BOSTANCI, Levent ERGÜDER, Serkan MACİT ve Ahmet SAYAR

Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü

Kocaeli Üniversitesi

{recep.bostanci1.08, levent.erguder1.08, serkan.macit1.08, ahmet.sayar}@kocaeli.edu.tr

**ÖZET**

Bu makale, servis odaklı mimariye (SOA) dayalı dağıtık görüntü işleme sistemini ortaya koymaktadır. Amacımız, görüntü işleme programlarını web servis olarak paketleyip, son kullanıcılar ve diğer dağıtık sistemler tarafından uzaktan erişilebilir hale getirmektir. Önerilen SOA mimarisi; Java Teknolojileri, Netbeans Web Servis kütüphaneleri ve Matlab kullanarak oluşturulmuştur. Görüntü işleme fonksiyonları, (Kenar bulma gibi) Matlab tarafından sunulur ve ilgili fonksiyonlar web servisler olarak paketlenir. Ayrıca, önerilen görüntü işleme web servislerinin kolay ve etkileşimli biçimde kullanılabilmesi için GUI geliştirilmiştir. Sistem sadece arayüze sahip son kullanıcılar tarafından değil, aynı zamanda diğer servisler tarafından da kullanılabilir. Önerilen sistemin etkinliği bazı test senaryolarıyla değerlendirilmiştir.

**Anahtar Sözcükler:** Dağıtık Sistemler, SOA, Web-servisler, Görüntü İşleme

**ABSTRACT**

This paper introduces a distributed image processing system based on service oriented architecture (SOA). Our aim is wrapping the image processing functionalities and programs as web-services and making them accessible by other distributed systems and end users. The proposed SOA architecture is built by using JAVA technologies, Axis2 Web-service libraries and MatlLab. Matlab provides the image processing functions (such as edge detection) and these are wrapped as Web-services. We also develop an easy to use, interactive GUI to use the proposed image processing web services seamlessly. The system can be accessed not only by the end users having GUI but also by other inter-operable services. The efficieny of the proposed technique is evaluated by some test cases.

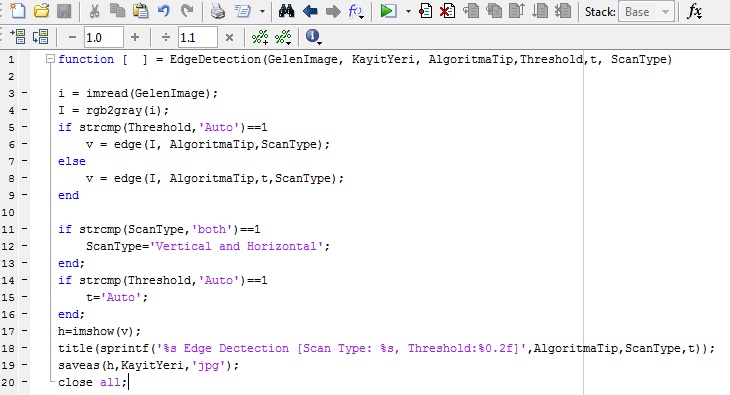
**Keywords:** Distributed Systems, SOA, Web-services, Image Processing

# GİRİŞ

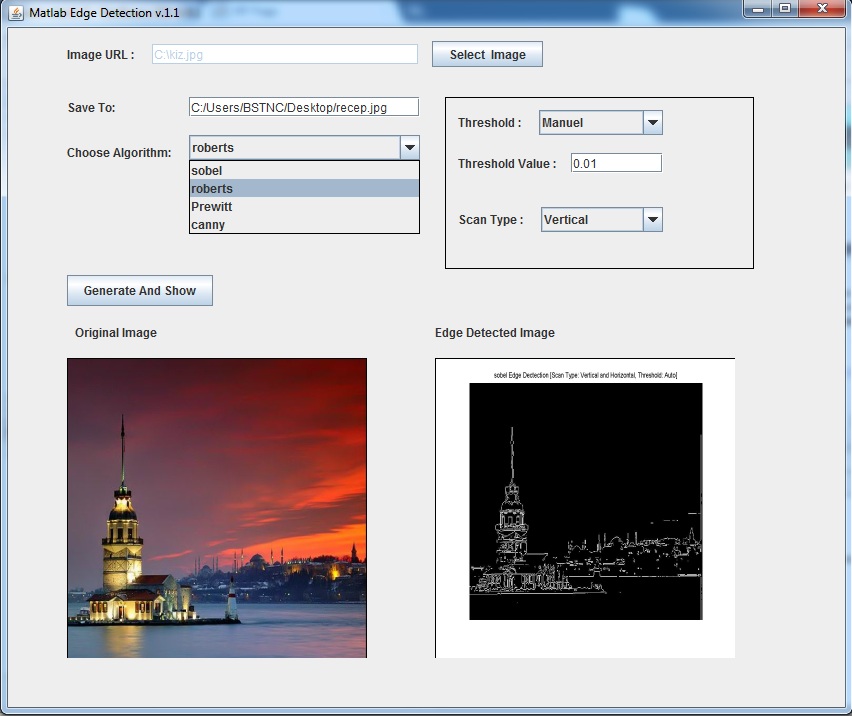
Sayısal bir görüntü bölgeler ve sınırlandıran kenarlardan oluşur. Görüntüdeki bir bölge genellikle aynı ya da benzer genlikleri paylaşan pikseller yığını olarak tanımlanabilen nesneleri gösterir. Kenar ise bir pikselden diğerine geçişteki piksellerin genliklerinde meydana gelen ani bir sıçrama olarak tanımlanır.(Güvenç ve Karagül, 2009). Başka bir ifadeyle, gri seviyeleri farklı iki bölge arasındaki geçiş veya sınır bölgesi kenar olarak belirlenir. Görüntülerden elde edilen kenar bilgileri sayısal görüntü işlemenin diğer analiz yöntemlerinde başarıyla uygulanmaktadır.(Güvenç,Karagül,2009)(Gonzalez, Woods,1993)

Görüntülerin kenarlarını belirlemek için araştırmacılar uzun bir süredir çalışmaktadır. Buna rağmen kesinleşmiş ve çok etkili bir kenar belirleme metodu bulunmamaktadır.(Albora ve Diğerleri,2007).Günümüzde kullanılan en popüler kenar belirleme yöntemleri Sobel, Prewitt, Robert ve Canny operatörleridir (Canny, 1986). Bu operatörler işlenecek olan pikselin yoğunluğu ile komşu piksellerin yoğunluklarının karşılaştırılması yöntemiyle çalışmaktadırlar. Ancak, bu operatörler görüntüdeki gürültüye çok duyarlı ve karmaşık matematiksel işlemler içermektedirler. Bu noktadaki karmaşıklığı azaltmak ve zamanı daha verimli kullanmak amacıyla Matlab vb. programlar kullanılmıştır ve bu sayede kenar belirleme ile diğer görüntü işleme operasyonlarını gerçeklemek oldukça kolaylaşmıştır.(McAndrew,2004)

Edge Detection (kenar bulma) algoritmaları görüntü işlemenin bir alanı olarak; plaka tanıma, yüz tanıma ve bulma, göz tanıma, harita ve uydu görüntüleme sistemlerinde aktif olarak kullanılmaktadır. Uygulamaya duyulan ihtiyaç göz önünde bulundurulduğunda hız, maliyet ve kalite de önemini artırmıştır. Matlab yazılımı kenar belirleme işlemini her ne kadar basit ve kısa bir komut setiyle uygulasa da, gerek hız gerekse maliyet açısından uygun olmayışı ve kullanıcıya daha basit bir arayüz sunmayışı bizi bu konudaki yetersizlikleri karşılayacak bir uygulama geliştirmeye yönlendirmiştir. Sonuçta ortaya konan uygulama Matlab’ın yaptığı edge detection işlemine uzaktan erişimli bilgisayar mimarisi, kullanıcı etkileşimli ve hızlı işlem görebilen bir yapı ekleyerek, bu işleme kayda değer bir katkı sağlamıştır. Edge Detection işleminin iki uygulama tarafından nasıl gerçekleştirildiğini daha somut kavrayabilmek için Şekil-1 ve Şekil-2 incelenebilir.



Şekil-1 Kenar belirleme için örnek bir Matlab uygulaması



Şekil-2 Kenar Belirleme için geliştirilen uygulama arayüzü

SOA, yani servis odaklı mimari, farklı platformların iletişiminin, entegrasyonunun sağlanması amacıyla bileşenlerin tek bir environment üzerinden tekrar kullanılabilen, standardize edilmiş ve birleştirilebilen servisler olarak diğer sistemlerin kullanımına açılmasıdır. Bir web servisi standardı olan SOA üzerinde, sözü edilen servisler tanımlanıp, konfigure edilerek dağıtık sistemlerin tek bir platform üzerinden, karşılıklı çalışabilir bir şekilde, birbirinden soyutlanmış olarak entegrasyonu sağlanır. Web servisleri, bağlantı ve iletişimle ilgili yöntemleri içerir. SOA ise tüm bir IT stratejisidir. Küçük projeler için uygun değildir, uygulamada başarısızlığa neden olabilir. Yapının geliştirilme amacı, büyük organizasyonların ihtiyacına cevap vermektir ( Computerworld Türkiye, 2010).

Web servisleri, SOA standardıyla çalışan uzaktan erişim mimari yöntemidir. Uygulamamız web servis client-server mantığında, ihtiyaç duyulan Matlab görüntü işleme metotlarını, belirlenen algoritmaları da kullanarak hizmete sunar. Kullanıcı, client host üzerinden istenilen paketi kullanarak,daha hızlı ve fonksiyonel olarak ihtiyacını giderebilir. Geliştirdiğimiz GUI, kullanım açısından basitliği ve etkileşimli yapısı ile kullanıcının herhangi bir kod yazmasına, bilgisayarına Matlab yazılımı kurmasına gerek kalmadan, işlemleri yürütmesini sağlar.

İlerleyen dönemde, web üzerinden ihtiyaca göre belirlenen başka paketler yayınlanabilir ya da yazılım geliştiricilerin uygulamalarında kullanabilmeleri için web servislerinin WSDL URL’leri yayınlanabilir.

# TEMEL BİLGİLER

Web servisleri 2000 li yılların başlarında ortaya çıkan ve bir çok yazılım firması tarafından yoğun bir destek bulan bir modeldir. Web servisleri açık internet standartlarına dayanır. Henüz gelişme ve olgunlaşma aşamasında olan bu modelle ilgili olarak bu aşamada ortaya çıkan ve kullanılan çekirdek standartlar SOAP, WSDL ve UDDI’dır.

Web Servisler uygulama bileşenidir ve açık protokolleri(open protocols) kullanarak haberleşir. XML temeline dayanırlar. Web servisler bilgisayar ve telekomünikasyon sektöründe internetten sonraki en büyük devrim olarak kabul edilir. Bu nedenle de büyük çaplı kurumlar bu konu üzerine yoğunlaşmış ve kendilerine uyarlamaya çalışmışlardır (Benslimane, Dustdar, Sheth, 2008).

Web servis mantığında, kullanıcı, diğer makinaya bir takım datalar gönderir ve geriye yine işlenmiş bir takım datalar alır. Diğer makinada gerçekleştirilen uygulamalar, kullanılan algoritmalar, fonksiyonlar hakkında hiçbir bilgisi yoktur. Web page mantığında ise, web sayfaları bir hostta yüklüdür. Kullanıcı, hostun çalışan dizinine bağlanır ve orada çalıştırılabilen sayfayı browser aracılığıyla görüntüler. Yani kullanıcı kendinde bir işlem gerçekleştirmez, her işlem host üzerindedir. Dolayısıyla güvenlik tehlikededir. Bu nedenle de web servisler daha hızlı,daha güvenli ve daha çok alanda etkileşimli olarak esnek kullanılabilir kabul edilir.

Uygulamamızdaki amaç da bu noktadan yola çıkarak, matematik dünyasını temel alan tüm mühendislik alanlarında, aktif olarak kullanılan Matlab yazılımının kullanımını, web sayfasından daha da etkileşimli kullanabilmeyi sağlamaktır.

Matlab programı da, command ekranından, farklı web sitelerinden verilen web servislerine erişip kullanabiliyor. Bunu *createClassFromWsdl*  foksiyonu ya da SOAP fonksiyonlarını kullanarak yapıyor. *createClassFromWsdl*  arka planda yine SOAP kullanır. Uygulama geliştiricisinin SOAP protokollerini bilmesine gerek kalmaz. Eğer WSDL bağlantısı yoksa, SOAP fonksiyonları(*createSoapMessage, callSoapService, and parseSoapResponse*) kullanmak zorunda kalır (The MathWorks, 2011).

Kurduğumuz örnek mimari (edge detection) veya isteğe göre yayınlayabileceğimiz paket ya da WSDL linkleri, kullanıcıların ihtiyaçlarını karşılamaya yönelik olacaktır. Plaka tanıma, göz tanıma, yüz tanıma sistemlerinde, image ilk olarak web servisinde işlendikten sonra, elde edilen yeni görüntüyü kullanıcı kendi algoritmasına da uygulayabilir.

Uygulamanın Java teknolojileri ile geliştirilmesinin amacı, açık kaynağın vermiş olduğu güven , hız ve farklı platformlara entegrasyonunun kolay olmasıdır. Kullanıcı bu sayede ister Linux, ister Windows ister MacOS ortamından servislere bağlanma imkanı bulabilir, kendi uygulamalarına entegre edebilir.

# UYGULAMA VE MİMARİ

Uygulama, Şekil 3 ‘te görüldüğü üzere, iki ana temel yapıdan oluşuyor. SOA Client(web service client) ve SOA Server (web service server). Veri akışı adım numaraları, transfer çizgileri üzerinde parantez içinde belirtilmiştir.

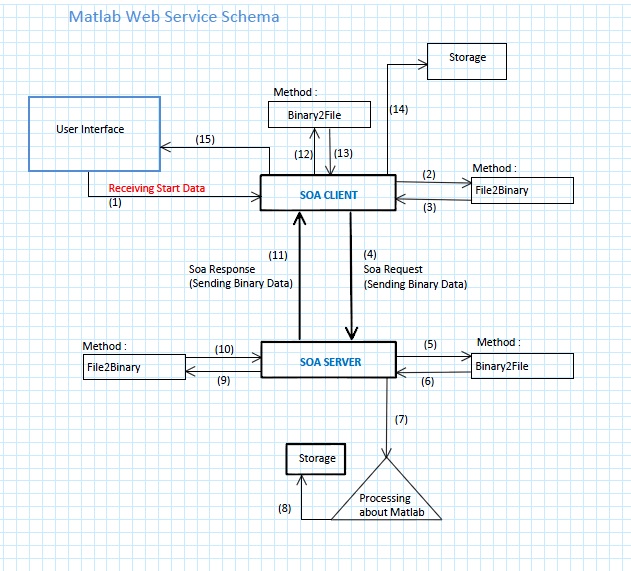
Kullanıcı kendisine sunulan paket aracılığıyla, GUI üzerinden, gerekli bilgileri girer, algoritma seçimini yapar ve Edge Detection uygulanacak olan resmi seçer. <*Process*> butonuna basıldığında, datalar işlenmek üzere, devamlı olarak erişilebilir ve run durumunda olan web service server’ına ulaşır. Server, gerekli işlemi kullanıcının seçimine göre yaptıktan sonra, geri döndürülecek image dosyasını client’a gönderir. Adım adım açıklaması ise;

*Adım 1.* Kullanıcıdan, GUI aracılığıyla bilgiler girmesi istenir. Bunlar kenar belirleme işlemi uygulanacak resim dosyasının seçilmesi, kullanılacak algoritmanın seçilmesi, üzerinde işlem uygulanan resmin hangi dizine hangi isimle kaydedileceği bilgileridir.

*Adım 2.* Seçilen resmin ağ üzerinden başka bir host’a gönderilebilmesi için binary array’e çevrilmesi gerekir. Bunun için metoda seçilen resmin URL’si gönderilir. Metod, resmi binary array’e çevirerek, diziyi *Adım.3* üzerinden tekrar main metoda gönderilir.

*Adım 4****.*** SOA Request olarak isimlendirilmiş adımdır. Web Service Server’a, kullanıcının GUI’de girdiği veriler ve seçilen resmin binary array’i gönderilir.

*Adım 5.* Ws-server, öncelikle binary array’i bir image file dönüştürmek için, gerekli metoda gönderir. Metot aldığı veriyi, server’da static olarak belirlenmiş bir dizine resim dosyası olarak kaydeder.*Adım 6* ile geriye resim dosyasının URL sini döndürür.



Şekil 3**:** Uygulamanın Genel Mimarisi

*Adım 7.* Matlab’ın Edge Detection işlemlerinin yapıldığı kütüphaneler daha öncesinden server uygulamasına dahil edilmiştir. Main metodumuz, kullanıcıdan aldığı işlenmemiş resmin static adresini, işlendikten sonra kaydedilecek resmin adresini ve kullanıcının tercih etmiş olduğu detection algoritmasını(sobel, roberts, prewitt), kendi yazdığımız library’den oluşturduğumuz sınıftaki metoda gönderir. Matlab bu noktada server makinasında arka planda çalışıyor durumdadır; ancak herhangi bir görsellik söz konusu değildir. Dolayısıyla herhangi bir yavaşlık, RAM üzerinde gereksiz alan işgali, CPU’yu gereksiz yere yorma gibi olumsuzluklar da söz konusu olmaz. “Matlab metodu” muz arka planda çalışıp, gerekli algoritmaya göre işlemleri yaptıktan sonra, yine developer tarafından önceden belirlenmiş static adrese kaydeder.(*Adım 8*) Bu işlenmiş resim dosyasının url’si main metod tarafından da biliniyordur.

*Adım 9.* Artık Server tarafında yapmamız gereken, işlenmiş olan resim dosyasını, client’a göndermektir. Bunun için yine binary array’e çevirme işlemine gerek duyuyoruz. File2Binary metodu geriye bir array döndürür.(*Adım 10*)

*Adım 11.* SOA-Response olarak adlandırdığımız adımdır. Kullanıcıya, işlenmiş resim dosyasının binary array’i gönderilir.

*Adım 12.* Ws-Client main metodu, server’dan aldığı binary array’i ve kullanıcıdan GUI aracılığıyla aldığı yeni işlenmiş resim adresini, Binary2File metoduna gönderir. Metod diziyi file’a çevirme ve kaydetme işlemlerini gerçekleştirir. *Adım 13, 14.*

*Adım 15.* İşlenmiş ve kaydetme işlemi başarıyla yapılmış resim, son olarak GUI ekranında, eorjinal resmin yanında görüntülenir.

# SONUÇ VE YORUMLAR

Uygulamanın farklılık ve üstünlükleri şu şekilde sıralanabilir;

1-Matlab uygulamasına ihtiyacı olan kullanıcı, gerekli paketi bilgisayarına indirerek fonksiyonları kullanabilir.

2-Kenar belirleme işlemine ihtiyacı olan uygulama geliştiricisi, gerekli WSDL URL’yi kullanarak yazılımına entegre edebilir.

3-Uygulamamız Java tabanlı olması dolayısıyla, multiplatform çalışma imkanına sahiptir. Hem Unix/Linux hem de Windows kullanıcıları çalıştırabilir. Ayrıca .Net ortamına da kolayca entegre edilebilir bir yapısı vardır.

4-Web Servis mantığında bir mimariye sahip olduğu için, yine farklı uygulamaların birbiriyle daha hızlı ve güvenli haberleşmesini kolaylaştırır. Bir masaüstü uygulaması ile web uygulaması projelerinin, data transferini mümkün hale getirir.

5-Matlab’ın bir fonksiyonuna ihtiyacı olan kullanıcı, kod öğrenmek zorunda kalmaz. Bir Image Processing işlemi, kendi projesi için ara işlem ise, geliştirdiğimiz mimariyi kullanması ona hız ve kolaylık getirir.

Yapılan çalışma halen sürdürülmektedir ve ilerleyen zamanda daha da geliştirilecektir. Gelen isteklere ve ihtiyaçlara göre yeni servisler geliştirilip, WSDL’leri ve kullanıcı uygulama paketleri bir web sitesi üzerinden yayınlanabilir. Paketlerin, masaüstü uygulamaları olabileceği gibi projelere entegre edilebilen küçük Java kütüphanesi de olması mümkündür. Bu paketlerin kullanımına, kullanıcı adı ve şifreli kullanımlar, extra güvenlik eklentileri de konulabilir. Paketlerin WSDL linkleri, bağlanılan web service server’ların IP adresi dinamik değiştirilebilir yapılabilir.

# KAYNAKÇA

Benslimane, D., Dustdar, S., Sheth, A. (2008). Services Mashups: *The New Generation of Web Applications*. *IEEE Internet Computing, vol. 12, no. 5*.

Canny, J. F. "A Computational Approach to Edge Detection", IEEE Trans Pattern Anal Mach Intel, 6, 679–698, 1986.

Computerworld Türkiye. (2010, January 4). SOA Tanımlamalar ve Çözümler. Service-Oriented Architecture (SOA) nedir? Retrieved November 27,2011, from <http://www.computerworld.com.tr/soa-tanimlamalar-ve-cozumler-detay_4327-sayfa_1.html>

Gonzalez, R. C., Woods, R. E. Digital Image Processing, Addison-Wesley, Reading, MA 1993

Güvenç, U., Karagül, T. (2009). Görüntülerin Kenar Haritalarının Çıkarımına Yeni Bir Yaklaşım.*BİLİŞİM TEKNOLOJİLERİ DERGİSİ*, C.2 S.1, s.23-27

McAndrew, A (2004). *An Introduction to Digital Image Processing with Matlab.* Retrieved November 30, 2011, from University of Eastern Finland, School of Computer Science and Mathematics Web site: <http://www.cs.joensuu.fi/pages/koles/image/book2511.pdf>

 The MathWorks, Inc.Matlab R2011b documentation. *Ways of Using Web Services in MATLAB.* Retrieved October 14, 2011, from <http://www.mathworks.com/help/techdoc/matlab_external/br0203q-1.html>