

.NET PLATFORMUNDA EMGU CV VE AForge.NET İLE GÖRÜNTÜ İŞLEME TEKNİKLERİNİN UYGULANMASI

Mehmet KARAKOÇ¹

¹Akdeniz Üniversitesi, Bilgisayar Bilimleri Araştırma ve Uygulama Merkezi (BAUM), 07058, Antalya, TÜRKİYE
mehmetkarakoc@akdeniz.edu.tr

Özet- Bu çalışmada, Görüntü İşleme (Gİ) tekniklerini uygulamak için kullanılan platformlar ve araçlar sınıflandırılarak kısaca tanıtılmıştır. Ayrıca Emgu CV ve AForge.NET kütüphanelerinin Gİ yetenekleri belirtilmiştir. Çalışma kapsamında, C# programlama dili kullanılarak bu kütüphanelere ait çok sayıda işlevi içeren bir program geliştirilmiş ve gözlemlenen sonuçlar sunulmuştur. Geliştirilen uygulamada, görüntü ve liste kutucukları ve filtreleme olayları aracılığıyla, görüntüler üzerinde temel filtreleme işlemleri gerçekleştirilebilmekte ve videolar işlenebilmektedir. Program, görüntü sıkıştırma, görüntülerden öznitelik çıkarımı, görüntü histogramlarını görüntüleme, iki görüntü arasında çeşitli işlemler uygulama ve videodan görüntüler veya görüntülerden video elde etmeye imkân tanımaktadır. Kütüphaneler, kapsam, esneklik ve kullanım kolaylığı açısından karşılaştırılmıştır. Geliştirilen program, kapsamlı uygulamalar için temel ve yardımcı bir araç olarak kullanılabilir. Tüm bu söz konusu işlemler, ilgili parametreler değiştirilerek kolaylıkla uygulanabilir.

Anahtar Kelimeler- Görüntü İşleme, .NET, Emgu CV, AForge.NET, C#.

APPLYING IMAGE PROCESSING TECHNIQUES WITH EMGU CV AND AForge.NET ON .NET PLATFORM

Abstract- In this study, the platforms and tools used for applying Image Processing (IP) techniques were briefly introduced by classifying them. Furthermore, the IP capabilities of Emgu CV and AForge.NET libraries were indicated. In the scope of this study, a program with a large number of functions from these libraries was developed by using C# programming language, and the results observed were presented. In the application developed, it is possible to perform basic filtering operations on images and process videos with the help of image and list boxes, and filtering events. The program allows to image compression, feature extraction from images, displaying the image histograms, applying various operations between two images and making images from a video or a video from images. The libraries were compared in terms of the scope, flexibility and ease of use. Developed program may be used as a basic and helpful tool for comprehensive applications. All of these aforementioned operations can easily be applied by changing the related parameters.

Keywords- Image Processing, .NET, Emgu CV, AForge.NET, C#.

1. GİRİŞ (INTRODUCTION)

Resimler ve videolar günlük yaşamın vazgeçilmez parçalarıdır. Görüntü, resim, grafik, çizim, video sahnesi vb. verilerin iki boyutta temsil edildiği bir veri türüdür. Bu veri, bir videoya ait sahne veya herhangi bir nesne için anlık görünüm olabilir ve $I(x, y)$ gibi bir fonksiyonla ifade

edilir. Burada, (x, y) konumunda (düzlem koordinatları) yer alan bir bilgi, bir resimdeki en küçük eleman olan piksel olarak adlandırılır. Her piksel için o noktadaki parlaklık ve renk bilgisi söz konusudur. RGB renk uzayı için kırmızı, yeşil ve mavi renk kanalları farklı tonlarda bir araya gelir. Bu değerlerin ayırık ve sonlu olduğu durumda (sayısal değerlerle temsil), veri sayısal görüntü olarak ifade edilir ve sayısal görüntü ikili (*binary*), gri tonlu veya renkli olabilir.

Görüntü İşleme (Gİ), sinyal işleme kapsamında yer alan ve sayısal görüntüler üzerinde amaca yönelik olarak birtakım işlemlerin gerçekleştirilmesi ile yeni görüntülerin elde edilmesini amaçlayan önemli bir çalışma alanıdır. Tıp, fizik, sanat, grafik, biyoloji, astronomi, biyomedikal, sağlık, robotik, hukuk, iletişim, mühendislik ve bilgisayar bilimleri gibi alanlarda ve güncel konularda uygulama alanı bulabilmekte ve teknolojik gelişmelerle birlikte hızlı bir biçimde gerçekleştirilebilmektedir. Gİ teknikleri, çeşitli platformlarda farklı araçlar aracılığıyla, çeşitli görüntüleme cihazları ve bilgisayar yazılımları yardımı ile görüntü içerikli verilerin kullanımı ve aktarımından, görüntü veri tabanları üzerinde görüntü içerikli sorgulamalara (imge erişimi) kadar geniş ölçüde (kuramsal çalışmalar ve pratik uygulamalar) uygulanabilmektedir.

Gİ için girdi olarak alınacak bilgi, herhangi bir alana ait resim, fotoğraf, ekran görüntüsü, video ve animasyon görüntüleri (çoklu ortam) gibi sayısal veri kaynakları veya ses/sinyal görüntüsüne ait görüntü verisi olabilir. Görüntüler fotoğraf makinesi, tarayıcı veya kamera gibi cihazlarla yakalanabilir. Bilgisayarlar aracılığı ile görüntülerden yeni bilgilerin elde edilmesi, bu verilerin işlenmesi, çözümlenmesi ve yorumlanması amaçlanır. Görüntüler üzerinde yapılan işlemler, noktasal (tek piksel), yerel (nokta ve komşuları, alt görüntü / blok) veya bütünsel (tüm görüntü) olabilir. Bu işlemler için matematik temelli (ikili ve aritmetik işlemler) veya görüntü histogramlarına bağlı çeşitli teknikler; yumuşatma, kenar belirleme gibi doğrusal; ortalama (*mean*), ortanca (*median*), “min”, “max” gibi doğrusal olmayan filtreler kullanılır. Filtre kullanımında, görüntü üzerinde bir filtre var gibi her pikselin değeri yeniden hesaplanır.

Görüntü yakalama/üretme, bulanıklaştırma, keskinleştirme (bulanıklığın azaltılması), sıkıştırma, onarma ve iyileştirme gibi uygulamalarda Gİ teknikleri kullanılır. Gürültü giderme, görüntü analizi, öznitelik çıkarımı, görüntüde metin gizleme ve (renkli) görüntüleme gibi amaçlarla, çeşitli çalışma alanlarına (örneğin yapay zekâ) ait yöntemler uygulanabilir. Görüntüleri karşılaştırmak için yapay sinir ağları, görüntü verilerinden anlamlı bilgi çıkarımı için veri madenciliği teknikleri tercih edilebilir. Bu kapsamdaki bazı uygulamalar ise şunlardır:

- Akıllı robot sistemleri, radar ve sonar sistemleri
- Biyometrik tanıma: Yüz (ifadesi/şekilleri), parmak izi, iris, el geometrisi, retina
- Coğrafi bilimler ve meteoroloji: Hava ve uydu görüntülerinden hava tahmini
- Doğal kaynakların izlenmesi, sel ve yangın kontrolü
- Endüstri: Süreç/ürün denetimi, kalite kontrol
- Görüntü temelli istatistiksel bilgi çıkarımı ve belge işleme
- Görüntü depolama, eski ve hasarlı fotoğrafların onarılması
- Hedef algılama ve tarama, uzaktan görüntü algılama, izleme uygulamaları
- Kenar tespiti: Görüntüdeki nesnelere/farklılıkların belirlenmesi
- Mayın tarama, arkeolojik öğelerden bilgi çıkarımı (arkeolojik kalıntıların tespiti)
- Nesne tanıma/takibi, hareket yakalama/takibi
- Oyun programlama: Bilgisayarlı görü (*computer vision*), 3-B modelleme
- Savunma sanayi: Gece görüşü, akıllı roket sistemleri
- Sınıflandırma: Resim görüntülerini (örneğin meyveler) özelliklerine göre gruplama
- Uzay bilimleri: Uzay görüntülerinden cisimlerin anlaşılması ve çözümlenmesi

Gİ tekniklerini esnek ve pratik bir şekilde uygulamak amacıyla kullanılacak çok sayıda (açık kaynak) kütüphane mevcuttur. Masaüstü ve web tabanlı ve mobil uyumlu uygulamalarda, pek çok disiplinde ve problemde Gİ ile karşılaşılır. Uygulamalar, bilgisayar kaynaklarının büyük ölçüde kullanımını ve verimli teknolojileri gerektirebilirler. Çözüm için hazır araçlardan ve

mevcut tekniklerden yararlanılabilir veya yeni algoritmalar geliştirilebilir. Bu çalışma kapsamında geliştirilen Gİ programı, Tablo 1’de verilen şu işlemleri yapmaya imkân tanır.

Tablo 1. Bu çalışma için gerçekleştirilen işlemler (Operations implemented for this study)

İşlem	Ayrıntılar
Resim İyileştirme	Filtreler: Gürültü giderme, görüntü iyileştirme, onarma ve düzenleme
Histogram İşlemleri	Histogram eşitleme ve görüntüleme
Dönüşüm İşlemleri	Çevirme ve döndürme işlemleri
Morfolojik İşlemler	Genişletme ve inceltme
Video İşleme	Sahne/çerçeve (<i>frame</i>) elde etme, videoya yazma, video üzerinde filtreleme işlemleri
Diğer Özellikler	Korelasyon, öznitelik çıkarımı, görüntüler arası işlemler, görüntü sıkıştırma
Kullanım Kolaylığı	C# ile GUI tasarımı, esneklik ve kolay kullanılabilirlik

Gİ amacıyla geliştirilen kütüphaneler ve araçlar aracılığıyla, görüntü temelli pek çok zor uygulama kolaylıkla gerçekleştirilebilir:

- Algılanan görüntü örüntülerini döndürme, ölçeklendirme, farklı renklerle ifade etme
- Güvenlik: Bilgisayar kontrollü güvenlik sistemleri ve personel takibi
- Görüntü iletimi ve depolama: Hava ve uydu görüntülerinin işlenmesi
- Görüntü temelli doğrulama ve/veya kimlik tespiti
- Görüntüler arasındaki farkların belirlenmesi, birleştirilmesi
- Resim, geometrik şekil, plaka ve karakter tanıma
- Tıbbi görüntülerden hastalık teşhisi: Kemiklerdeki kırıkların belirlenmesi, damar tıkanıklığının anlaşılması, sağlıksal görüntü taramalarının işlenmesi

Gİ gerektiren uygulamalarda, istatistiksel, yapısal veya zeki yollar tercih edilebilir. Bu kapsamdaki en önemli adımlardan biri mevcut görüntülerin iyileştirilmesi, bir diğeri ise bu görüntülerden kaliteli özniteliklerin (örüntüye ait tanımlayıcılar) çıkarımıdır. Çevresel ışıklandırma durumlarına (ortam yeterince aydın olmayabilir) ve görüntü yakalama araçlarına bağlı olarak; yakalanan görüntü gürültülü, hasarlı veya kalitesiz olabilir. Bu nedenle, ön işleme kapsamında, görüntü daha belirgin/keskin hâle getirilebilir ve onarılabilir. Görüntüleri iyileştirmek için gürültüleri giderilebilir veya görüntülere renk dönüşümü uygulanabilir. Görüntülerin çözünürlükleri düşürülerek, algoritmaların işletimi belirgin ölçüde hızlandırılabilir.

1.1. Literatür Taraması (Literature Review)

Bir resmin sayısal ortamlarda görüntülenmesi veya parlaklaştırılmasından, görüntü veri tabanlarında görüntü içerikli sorgulama yapma veya bir görüntüde yer alan şekle benzer şekilleri içeren başka görüntüleri aramaya kadar basit-zor pek çok uygulama birer Gİ problemidir.

İmgelerde, bir kısım dönüşüme karşı değişmeyen özel bölgeler, çok sayıda bilgisayarlı görü uygulamasında kullanılır. Güney ve Arıca [1], bu ilgi bölgelerinin üst seviye imge özniteliklerine dayalı olarak tespit edilmesi için “maksimum durağan uç bölgeler” yaklaşımını, imge noktalarının desen bilgisini kullanarak genişletmişlerdir. Ayrıca farklı ölçek ve yönelimdeki süzgeçler ile imgenin evrişimiyle (*convolution*) elde edilen çok boyutlu nokta öznitelikleri kullanılarak, etrafından daha farklı desene sahip imge bölgeleri (*blob*) çıkarılmıştır.

Bir sahnenin üç boyutlu yapısını elde edebilmek için farklı bakış açılarından çekilmiş stereo görüntüleri üzerindeki her bir noktanın uzaydaki koordinatlarının elde edilmesi, yani diğeri görüntü üzerindeki eşinin bulunması gereklidir. Stereo görüşte birden fazla kamera ile farklı bakış açılarından alınan görüntüler kullanılarak, üç boyutlu sahne çıkarımı yapılabilir. Özcan ve diğ. [2], stereo görüntülerden sahnedeki nesnelerin üç boyutlu derinlik kestirimlerinin nasıl yapılabileceği üzerinde durmuşlardır. Stereo eşleme işlemi için OpenCV’de [3] tanımlı blok eşleme ve yarı-bütünsel blok eşleme algoritmalarını kullanmışlardır. Amaç, bir piksel ve komşu

pikselleri ile blok hâlinde bir başka resimde arama yapma ve eşini bulmadır. Doğan ve diğ. [4] ise kuş bakışı görüntü dönüşümünü OpenCV ile gerçekleştirmişlerdir.

Gİ teknikleri, Çomak ve diğ. [5] tarafından inşaat mühendisliği alanında beton teknolojilerinde; Aksu ve diğ. [6] tarafından bulut bilişim teknolojisi ile web tabanlı bir biçimde uygulanmıştır. Varlık ve Çorumluoğlu [7], biyometrik kimlik doğrulama amacıyla, kişiye ait parmak izi ve yüz bilgilerinden yararlanmışlardır. Bu kapsamda, parmak izi ile kişi tanıma ve yüz ile doğrulama işlemleri gerçekleştirilmiştir.

Emgu CV [8] ve AForge.NET [9] ile ilgili bir başka inceleme olarak, Toth'un [10] çalışmasına başvurulabilir. Grycuk ve diğ. [11], görüntüdeki nesnelere çıkarılması (*blob detection*) amacıyla, görüntü bölütleme için yeni bir yaklaşım önermişlerdir. Önerdikleri yaklaşım, sadece görüntüdeki belirgin elemanları seçerken gereksiz noktaları eler. Görüntüde ilgilenilen nesnelere elde etmek için hem AForge.NET hem de Emgu CV kullanılmıştır. Liao ve diğ. [12], kameralı bir ortamda nesne izlemeyi ele almışlar; PTZ kamera için aynı nesnenin aynı şekilde etiketlenmesini (tutarlılık sağlanmalı) amaçlamışlardır. Bu işlemi gerçeklemek için bir yaklaşım önermişler, bu iki kütüphaneyi kullanarak bir prototip gerçekleştirmişlerdir.

Çalışmanın geri kalan kısmı şu şekilde düzenlenmiştir. Gİ araçları ve ilgili çalışmalar 2. bölümde, çalışma kapsamında Emgu CV ve AForge.NET ile gerçekleştirilen çok sayıda uygulama ise açıklamalı olarak 3. bölümde verilmiştir. Sonuçlar ve tartışma ise 4. bölümde yer almaktadır.

2. YÖNTEM (METHOD)

Gİ araçları, temel resim işleme algoritmaları ile sınıflandırma, grafiksel kullanıcı arayüzleri sunma, görüntüler ve videolar üzerinde düzenleme yapabilmek, bir nesnenin ana hatlarını bulma, şablon/şekil eşleştirme, yüz tanıma ve hareket izleme gibi pek çok imkân sunar. Kütüphaneler içinde tanımlı düşük ve yüksek seviyeli algoritmalar ile kaydetme, yükleme ve renksel değişiklik yapma işlemlerinden, dokusal ve şekilsel işlemlere kadar pek çok uygulama gerçekleştirilebilir. Gİ uygulamak için SimpleCV, GNU Octave, AForge.NET, VTK, FIJI, OpenGL, GIMP (*GNU Image Manipulation Program*), Inkscape ve Blender gibi araçlar kullanılabilir. Ayrıca web tabanlı servislerden masaüstü uygulamalara kadar pek çok araç da mevcuttur. Gİ uygulamaları, C++, Java ve Python gibi programlama dilleri ile uygulanabilir.

Açık kaynak bir bilgisayarlı görü kütüphanesi olan OpenCV (*Open Source Computer Vision*: <http://sourceforge.net/projects/opencvlibrary/>), bir resimden/videodan bilgiler çıkartıp işleyebilmek için INTEL tarafından C ve C++ dilleri ile geliştirilmiştir. Birçok algoritmaya sahip, modern programlama dilleri ile kullanıma imkân tanıyan, esnek, kolay ulaşılabilir ve hızlı bir kütüphanedir. Yaygın olarak kullanılır, platformdan bağımsızdır ve gömülü sistemlerde kullanılabilir. C fonksiyonları ve C++ metodları doğrudan C#'tan çağrılmayacağı için kütüphanenin .NET dillerinden biri ile kullanılabilmesi amacıyla çeşitli metodlar gereklidir. Bu nedenle, OpenCV'ye ulaşmayı sağlayan, C# programlama dili ile yazılmış bazı ara kütüphaneler (*wrappers*: Emgu CV, OpenCVSharp [13], OpenCVDotNet [14]) hazırlanmıştır.

2.1. Emgu CV

Emgu CV, OpenCV çatısı için platformlar arası bir .NET aracıdır. .NET uyumlu C#, VB.NET, VC++, C++, IronPython vb. dillerde OpenCV fonksiyonlarını doğrudan kullanabilmek imkânı tanımaktadır. Mono'da derlenebilmekte ve Windows, Linux, Mac OS X ve popüler mobil platformlar (iPhone, iPod Touch, iPad ve Android cihazları, iOS) üzerinde çalıştırılabilmektedir [8]. Güvenli olmayan (*unsafe*) kod kullanılmadan ve tamamen C# ile yazılmış bu kütüphane, C#

geliştiricilerine özellikle gerçek-zamanlı (*real-time*) bilgisayarlı görü uygulamaları geliştirme imkânı tanımakta ve iyi bir başarıyı sağlamaktadır.

Emgu CV, otomatik çöp toplama (*automatic garbage collection* - bellek yönetimi), XML desteği, çizim fonksiyonları, genel ve kapsamlı işlemler gibi avantajlar; çok sayıda açık kaynak örnek uygulama ve iyi bir belgeleme içermektedir. Geniş ölçüde Gİ algoritmaları mevcuttur ve genel görüntü sınıfı ile farklı renk ve derinlik birleşimlerinde çalışılabilir. Örnek uygulamalara, <http://www.emgu.com/wiki/index.php/Tutorial> bağlantısında yer alan “Examples” bölümünden ulaşılabilir. Emgu CV için en belirgin avantajlar ve dezavantajlar Tablo 2’de verilmiştir.

Tablo 2. Emgu CV için belirgin noktalar (Significant points for Emgu CV)

Avantaj	Dezavantaj
Ücretsiz (akademik ve ticari kullanım), açık kaynak, çoklu platform desteği	Algoritmaların karmaşıklığı ve derinliği, anlama ve kullanma için daha fazla zaman gerektirir.
Tam olarak bilgisayarlı görü odaklıdır.	
Kütüphane temelinde C programlama dilinde yazılmıştır: Optimize edilebilir ve Assembly seviyesine daha yakındır.	

Gade ve Moeslund [15], bir spor oyununa ait termal video için gerçek-zamanlı çevrim-içi izleme algoritması sunmuşlardır. İzlemeyi, Kalman filtresini temel alarak ve Emgu CV ile gerçekleştirmişlerdir. Philip ve Omotosho [16] ise gürültü giderme, nesne tanımlama ve öznelik çıkarımı için önerilen teknikleri Emgu CV ile gerçekleştirmişlerdir. Tayade ve diğ. [17], sürücünün uyuklama hâlini belirlemek için göz durumunu izlemişler; gerçekleştirim için Emgu CV kullanmışlardır. Meena ve Raja [18], temel k-ortalama algoritmasını, AForge.NET çatısı ve MATLAB çalışma ortamı ile sınımışlar ve bölütleme işlemi sonrasında elde edilen görüntüleri karşılaştırmışlardır. Rygał ve diğ. [19], görsel bilginin anlamsal bir biçime dönüştürülmesi için yeni bir algoritma sunmuşlar; Gİ amacıyla Emgu CV kullanmışlardır.

2.2. AForge.NET

AForge.NET, bilgisayarlı görü ve yapay zekâ alanlarındaki geliştirici ve araştırmacılar için tasarlanmış bir C# çatısıdır [9]. Ücretsiz ve açık kaynak bu çatı, Gİ, bilgisayarlı görü, evrimsel programlama (örneğin genetik algoritmalar), sinir ağları, bulanık hesaplamalar, yapay öğrenme (makine öğrenmesi), robotik (bazı araçlar mevcuttur) ve video işleme gibi bir dizi kütüphane ve örnek uygulamalar içermektedir. Bu çalışma için geliştirilen programda görüntüleme için kullanılmış, Gİ kapsamındaki rutinlerden ve filtrelerden ve ayrıca video bileşenlerinden yararlanılmıştır. AForge.NET için en belirgin avantajlar ve dezavantajlar Tablo 3’te verilmiştir.

Tablo 3. AForge.NET için belirgin noktalar (Significant points for AForge.NET)

Avantaj	Dezavantaj
Gİ algoritmaları için basit fonksiyonlar mevcuttur.	Önde gelen resmi bir kurum tarafından geliştirilmemiştir.
Pek çok örnek uygulamaya erişilebilir.	Bütün olarak bilgisayarlı görü odaklı değildir.
Kullanıldığı projeler mevcuttur.	Yalnızca Windows desteklenmektedir.

Köksal ve Şaykol [20], kameralar ile hareketli nesnelere gerçek-zamanlı bir biçimde algılamak ve farklılıkları tespit etmek için AForge.NET çatısında yer alan Gİ kütüphanesi dışında “Vision.Motion” kütüphanesini kullanmışlardır. Çalışmalarında, videoda arka arkaya gelen sahneler arasındaki farklar bulunur ve hareket farklılıkları tespit edilir. Ahmed ve diğ. 0, el hareketi tanıma için herhangi bir web kamerasından sahne yakalamak amacıyla, AForge.NET çatısında yer alan “Video” kütüphanesinden yararlanmışlar; el izleme yazılımı tasarlamışlardır.

2.3. Görüntü Temelli Çalışmalar (Image Based Studies)

Çeşitli araçlar kullanılarak, bilgisayarlı görü ve Gİ algoritmaları ile şu çalışmalar yapılabilir:

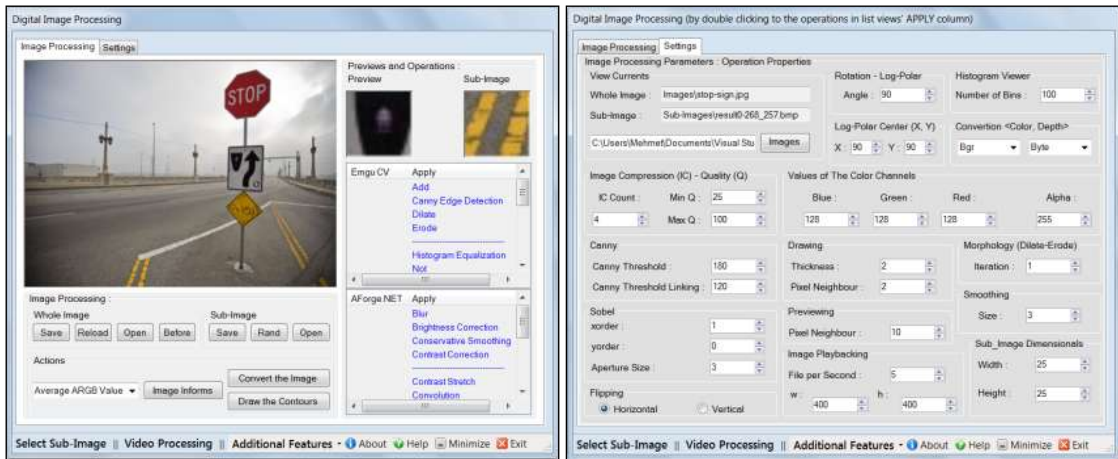
- İnsan-bilgisayar etkileşimi (*Human-computer interaction*)

- Nesne tanımlama, bölütleme (*Object identification, segmentation*)
- Yüz, işaret dili tanıma (*Face, gesture recognition*)
- Hareket yakalama, algılama ve takibi (*Motion capturing, detection and tracking*)
- Çiftli ve çoklu kamera ölçümlene ve derinlik hesaplama (*Stereo and multi camera calibration, and depth computation*)
- Hareketli/gezgin robot teknolojileri (*Mobile robotics*)

Örneğin yüz tanıma için yüze ait karakteristik özellikler analiz edilerek, ilgili noktalar ve bölgeler elde edilip veri tabanında tutulabilir. Tanınacak bir yüz verisinden çıkarılan benzer bilgiler (betimleyici vektörler) ise veri tabanındaki özellik vektörleri (referans betimleyiciler) ile karşılaştırılarak eşleştirilirler. En benzer/yakın görüntüler elde edilerek, ilgili yüze ait kişiler belirlenebilir. Ayrıca görüntüler üzerindeki ilgili olunan bölgeler de tespit edilebilir. Personel/öğrenciler/üyeler için devam durumlarının takibi, kontrol geçiş sistemleri veya eğitim-kamu kurum ve kuruluşlarında benzer uygulamalar için Gİ araçlarına ihtiyaç duyulur.

3. BULGULAR (FINDINGS)

Görüntüler, iyi bir şekilde yorumlanabilme, çözümlenebilme ve elde edilecek sonuçlardaki hata payının düşürülebilmesi için üzerinde işlem yapmadan önce ön işlemeye tabi tutulurlar. Örneğin orijinal görüntünün daha uygun bir biçimde görüntülenebilmesi için görüntü iyileştirme teknikleri tercih edilebilir. Bu kapsamda geliştirilen programın arayüzü Şekil 1’de verilmiştir.



Şekil 1. Geliştirilen programın arayüzü (Interface of the developed program)

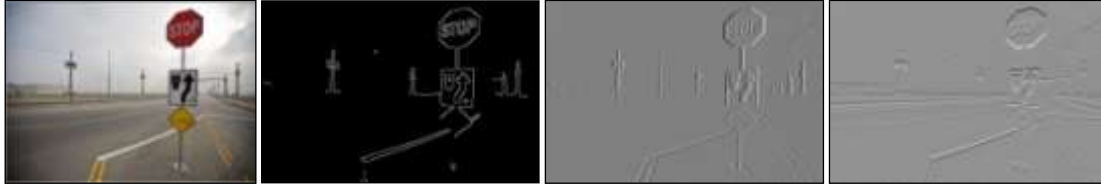
3.1. Resim İyileştirme İşlemleri (Image Enhancement Operations)

Emgu CV: *Add* ile görüntünün her renk kanalına belli değerler eklenir (Şekil 2(b) - mavi, yeşil, kırmızı, alfa renk kanalları için değerler sırasıyla 128, 128, 128 ve 255 (tam parlak)). *Add* işlemi iki görüntü ile de uygulanabilir. Bir resimdeki bazı öğelerin ayırt edilebilmesini kolaylaştırmak için ise resme *değil* işlemi uygulanır. *Not* renk değerlerinin tersini alır (Şekil 2(c)).



Şekil 2. Ekleme (b) ve tersini alma (c) işlemleri (Adding (b) and inversion (c) operations)

Kenarların belirlenmesi ile görüntüdeki renk geçişlerinin keskin şekle getirilmesi mümkün olur (Şekil 3(b)). Ayrıca nesnelerin ve/veya farklılıkların elde edilmesi de sağlanmış olur. “Sobel”, görüntüdeki kenarların tespit edilmesi için kullanılan yöntemlerden biridir. “xorder=1, yorder=0, aperture_size=3” olarak uygulanması Şekil 3(c)’de, “xorder=0, yorder=1, aperture_size=3” olarak uygulanması ise Şekil 3(d)’de verilmiştir. Kenar belirleme algoritmaları ile bir görüntüdeki köşegensel kenarlar da belirlenebilir.



Şekil 3. Canny uygulama (b) ve görüntüdeki dikey (c) ve yatay (d) kenarların tespit edilmesi (Applying Canny (b) and detection of the vertical (c) and horizontal (d) edges in the image)

Şekil 4(b)’de “128 (mavi renk kanalı), 255, 255, 255” değerleri ile eşikleme işlemi verilmiştir.



Şekil 4. Eşikleme işlemi (b) (Thresholding operation (b))

Görüntü eşikleme, renkli veya gri tonlu görüntülerde uygulanarak, piksel değerlerini çok büyük/küçük değerlere filtreler. Amaç, bir görüntüdeki ön ve arka planların probleme yönelik olarak ayrıştırılması veya ilgili olunan bölgelerin veya herhangi bir nesnenin belirlenmesidir. Piksel değerlerinin belirlenen eşik değerinden büyük olduğu durumlarda “beyaz” ve diğer durumlarda “siyah” olarak belirlenmesi ile siyah-beyaz bir görüntü elde edilebilir. Eşikleme, görüntünün tümü yerine görüntü parçaları üzerinde uygulanarak *bölütleme* gerçekleştirilebilir.

AForge.NET: *Blur* görüntüyü bulanıklaştırırken, *Sharpen* keskinleştirir (Şekil 5). Bulanık görüntülerin keskinleştirilmesi ile göreceli piksel zıtlığı (kontrast) artar ve bulanıklık azalır.



Şekil 5. Bulanıklaştırma (b) ve keskinleştirme (c) (Blurring (b) and sharpening (c))

“BrightnessCorrection” ile parlaklık, “ContrastCorrection” ile zıtlık ayarlanır (Şekil 6).



Şekil 6. Parlaklık (b) ve zıtlık (c) ayarlama (Brightness (b) and contrast (c) adjustments)

Görüntü üzerindeki gürültüyü gidermek için ortalama ve ortanca filtreler kullanılabilir (Şekil 7).



Şekil 7. Ortalama (b) ve ortanca (c) filtreler (Mean (b) and median (c) filters)

Ortanca filtre (kendi ve etrafındaki pikseller arasında ortadaki değer alınır), resmi ortalama filtreye (kendi ve etrafındaki piksellerin ortalama değeri alınır) göre daha az bulanıklaştırır. “Salt and Pepper Noise” ile görüntüye gürültü ekleme (AForge.NET), “Smooth Median” ile gürültülü resmin yumuşatılarak gürültülerinin giderilmesi (Emgu CV) ve “Conservative Smoothing” ile gürültülü resmin yumuşatılması (AForge.NET) işlemleri Şekil 8’de verilmiştir.



Şekil 8. Gürültü ekleme (b), gürültü giderme (c) ve resmi yumuşatma (d) (Adding noise (b), removing the noise (c) and smoothing the image (d))

Gürültü giderilirken ve görüntü yumuşatılırken, gürültü ile birlikte bazı ayrıntılar yok olabilir.

3.2. Çok Görüntülü İşlemler (Operations with Multiple Images)

Emgu CV: *And* iki görüntü arasında mantıksal “ve” işlemini gerçekleştirir. *Comparing* ise iki görüntüyü birbiri ile karşılaştırmak için kullanılır ve bir karşılaştırma maskesi döner. *Equals* görüntüleri birbiri ile karşılaştırır; tüm pikseller aynı ise *true*, değil ise *false* döner. *Max* ve *Min* ise ilgili piksel değerleri için sırasıyla en büyük ve en küçük değerleri alır.

AForge.NET: İki görüntü için en küçük pikseller *Intersect*, en büyük pikseller *Merge* ve fark bilgisi ise *Difference* fonksiyonları ile elde edilir.

3.3. Emgu CV ile Histogram İşlemleri (Histogram Operations with Emgu CV)

Görüntüdeki renk aralıklarını genişletmek amacıyla uygulanan histogram eşitleme, “EqualizeHist” metodu ile gerçekleştirilir (Şekil 9(b)). Histogram bilgisi ise “HistogramViewer” ile görüntülenir. Şekil 9(c)’de, sırasıyla görüntünün mavi, yeşil, kırmızı ve alfa renk kanalları için histogramları görülmektedir. Histogramlar, resmin piksel değerlerinin yoğunluklarını grafiksel olarak gösteren şekillerdir. Görüntünün zıtlığı hakkında bilgi verirler ve yatay ve dikey eksenler boyunca sırasıyla parlaklık ve yoğunluk bilgileri görülür. Nesnelerin çıkarılması için histogramlardan yararlanılabilir.



Şekil 9. Histogram eşitleme (b) ve histogram bilgisini görüntüleme (c) (Histogram equalization (b) and displaying the histogram information (c))

3.4. Dönüşüm İşlemleri (Transformation Operations)

Bu alt bölümde, mevcut görüntülerden, farklı ölçeklendirme, boyutlandırma, renk uzayları (RGB-YCrCb-HSV-HSL) ve değerleri ile yeni görüntüler elde edilmesi (Emgu CV metotları kullanılmıştır) yer almaktadır. Örneğin *Flip* komutu ile görüntü yatay veya dikey olarak çevrilirken, *Rotate* görüntüyü belirlenen açı değeri kadar döndürür (Şekil 10).



Şekil 10. Görüntünün yatay (b) ve dikey (c) olarak çevrilmesi ve saat yönünde 90 derece döndürme (d) (Flipping the image horizontally (b) and vertically (c), and rotating it (d) clockwise by 90 degrees)

3.5. Morfolojik İşlemler (Morphological Operations)

Görüntülere (doğal/yapay) ait yüzeylerin morfolojik birer yapısı vardır. Bu yapılar üzerinde farklı bilim dallarında çalışmalar yapılır. Bir görüntüye ait yüzeyin görünümü, görsel doku (*visual texture*) olarak adlandırılır ve farklı yoğunluk birleşimleri içerebilir. Bu oluşumlar pürüzlük gibi fiziksel sonuçlara neden olabilir. Yerel bölgelerde yoğunluk tek düze olabilir; fakat görüntü genelinde bu şekilde olmaması, dokusal-şekilsel işlemleri karmaşıklaştırabilir.

Geometrik fonksiyonlar, bir görüntüyü genişletme, daraltma, eğme ve döndürme için uygulanır. Bir görüntüyü eğme/döndürme işlemiyle, görüntüler üst üste yerleştirilebilir veya nesne tanıma için kullanılan görüntülerin eğitim kümeleri yapay olarak genişletilebilir. Geometrik dönüşümler sonucunda görüntü içeriği değişmez, fakat yapılandırma elemanları (birek matris) kullanıldığı için piksel ızgaralarının biçimleri bozulur. Renkli görüntüler için her renk kanalı bağımsız olarak işlenir. Genişletme/büyütme (“dilata”) ve aşındırma/bozma (“erode”) işlemlerinin görüntü üzerinde belirli yineleme sayısı kadar uygulanması (Emgu CV metotları kullanılmıştır) Şekil 11’de verilmiştir.



Şekil 11. Genişletme (b) ve aşındırma (c) işlemlerini ikişer kez uygulama (Applying dilatation (b) and erosion (c) operations twice per each)

İkili görüntülerde erozyon işlemi beyaz alanları yumuşatmak amacıyla kullanılır. Giriş pikselinin komşu piksellerinden en az biri 0 değerine sahipse, çıkış pikselinin değerine 0 atanır. Genişletme işlemi siyah alanları yumuşatmak amacıyla kullanılır. Giriş pikselinin komşu piksellerinden en az biri 1 değerine sahipse, çıkış pikselinin değerine 1 atanır. İkili görüntüye sırasıyla erozyon ve genişletme filtreleri uygulanması açma (*opening*), genişletme ve erozyon filtreleri uygulanması ise kapama (*closing*) olarak tanımlanır. Kapama çıktısı genişletme çıktısına benzemekle birlikte, beyaz alanlar kapama işleminde daha fazla aşınır.

3.6. Video İşleme (Video Processing)

Video işleme kapsamında, Emgu CV'nin "VideoSurveillance" uygulamasından yararlanılmış; iki ayrı uygulama geliştirilmiştir:

1. Bir videodan görüntüler elde etme
2. Bir klasörde yer alan resimleri birer sahne olarak video formatında yazdırma ("VideoWriter" kullanılmıştır)

Kamera ile elde edilen video görüntüler üzerinde, Emgu CV'nin görüntü kutucukları yardımıyla çeşitli işlemler uygulanabilir. Video işleme aracına ait arayüz Şekil 12(a)'da verilmiştir.

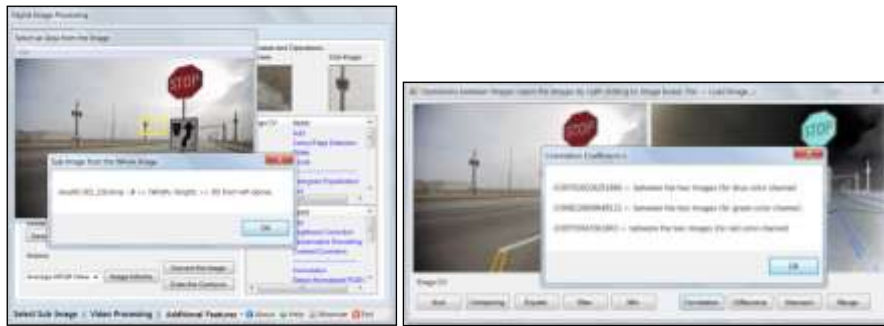


Şekil 12. Video işleme arayüzü (a) ve video sahneleri üzerinde uygulanan işlemler (b-c) (Video processing interface (a) and the operations applied on video scenes (b-c))

Tüm video taranabilir (Şekil 12(b)), seçili videodan anlık (bir önceki/sonraki) sahnelere ait çeşitli resimler görüntülenebilir (Şekil 12(c)) ve tüm sahneler veya ilgili sahne kaydedilebilir.

3.7. Ek Özellikler (Additional Features)

Bu kapsamda, alt görüntü seçme için görüntünün herhangi bir kısmını keserek ayırma/kırpma (Emgu CV: GetSubRect) ve aynı boyutlardaki iki görüntü arasında, her renk kanalı için ilişki değerini ve yönünü ölçme (AForge.NET: Correlation) amacıyla korelasyon uygulama mevcuttur (Şekil 13).



Şekil 13. İki görüntü arasındaki işlemler (Operations between two images)

Geliştirilen programla, bir görüntüye belli bir kalite aralığında belirli sayıda sıkıştırma işlemi uygulanabilir. Elde edilen görüntüler ile bir video oluşturularak, sıkıştırılmış görüntüler

arasındaki deęişim geişlerle izlenebilir; görüntü kalitesi ve boyutu üzerinde deęerlendirmeler yapılabilir. Ayrıca, siyah-beyaz görüntüde siyah olmayan piksel sayısı (“Count Non Zero”), görüntüdeki konturlar, her renk kanalı için görüntüdeki piksellere ait ortalama ve standart sapma deęerleri gibi görüntü öznitelikleri çıkarılarak, birtakım bilgiler de elde edilebilir. Bir sahne, farklı geometrik ve fotometrik şartlar ve deęişik görüntüleme şartlarında (farklı büyüklük ve dönüklük durumları) tespit edilmek; benzer veya aynı renge sahip bölgeler (komşu noktalar) ilişkilendirilmek veya ayırt edici sınırlara sahip homojen kısımlar çıkarılmak istenebilir (görüntü bölütleme). Şekil 14’te yer alan görüntüde şu bilgiler verilmektedir:

- Sol üst : Orijinal görüntü
- Sağ üst : Elde edilen şekillerin dış hatlarını ve biçimini ifade eden koordinatlar
- Sol alt : Bu şekillerin sınırlarını ifade eden dikdörtgenler
- Sağ alt : İlgili dikdörtgenler, en küçük alana sahip dikdörtgenler olacak şekilde



Şekil 14. Kapsamlı uygulamalar için şekillere ait niteliklerin gösterimi (The representation of the shapes features for comprehensive applications)

Bir görüntüde yer alan nesneye ait şeklin belirlenmesi karmaşık bir uygulamadır. Bu işlem, bu bölümde verilen uygulamalar şu şekilde uygulanarak gerçekleştirilebilir:

- Görüntü yakalama: Bir nesnenin anlık görünümü alınır.
- Gri tonlu görüntüye dönüştürme: Renkli görüntüye göre daha az yer kaplar.
- Görüntülerdeki gürültüleri azaltma: Ortanca/ortalama filtre, histogram eşitleme
- Kenar bilgisi: Kenar belirleme algoritmaları ile elde edilir.
- Şekil hatları: Kenar bilgisi şekil bulma metodunda kullanılarak bulunur.
- Görüntüdeki cisimlere ait şekil hatları, çizim metodu ile istenilen renkte gösterilir.

“PyrDown” ile daha düşüğe ve “PyrUp” ile daha yükseğe örnekleme yapılabilir (Emgu CV). Görüntünün yağlı boya çalışması olarak sunumu için “OilPainting” (AForge.NET) kullanılır. Görüntüden normalleştirilmiş renk kanalının çıkarılması için AForge.NET Gİ metodlarından biri olan “extract normalized RGB channel” işlemi kullanılabilir. Görüntüyü, her eleman görüntünün tek bir renk kanalını sunacak şekilde gri tonlu görüntüler olarak bir diziye aktarmak amacıyla da Emgu CV Gİ metodlarından “Split” kullanılır.

4. SONUÇ VE TARTIŞMA (CONCLUSION AND DISCUSSION)

Bu çalışmada, Emgu CV ve AForge.NET kütüphanelerinin Gİ kapsamındaki yetenekleri ele alınmıştır. Bu amaçla, C# programlama dilinde bir uygulama geliştirilmiş ve Gİ teknikleri,

esneklik, kullanım kolaylığı ve görsel arayüzler açısından kütüphanelere ait birtakım bilgiler verilmiştir (genel karşılaştırma Tablo 4'te). Emgu CV'nin güçlü ve gürbüz bir araç olduğu aşikârdır. Geliştiricinin C ve C++ dillerini daha iyi bildiği durumlar için Emgu CV kullanımını doğal olarak daha iyi bir seçim olacaktır (OpenCV nedeni ile). Bu noktada, geliştiricilerin yetenekleri ve deneyimleri farklılıklar gösterebilir. .NET platformunda Gİ yazılımları geliştirecek kişiler için Emgu CV oldukça uygun, fakat daha karmaşık bir araçtır. Basitlik ve bir proje için gerekebilecek destek açısından bakıldığında ise AForge.NET oldukça uygundur.

Tablo 4. Kütüphaneler için bir karşılaştırma (A comparison for the libraries)

Ölçüt	Emgu CV	AForge.NET
Özellik	OpenCV için ara kütüphane	Salt .NET kütüphanesi
Örnek Uygulama	Az Sayıda: Yeterli uygulama mevcut değil (OpenCV olarak bakılmalı)	Çok Sayıda: Kütüphanenin kullanımını gösteren çok sayıda uygulama mevcut (pek çok Gİ tekniği)
Anlama	OpenCV'yi anlama gereği	Kolay anlaşılır
Belgeleme	İyi	Zayıf
Kullanıcı Dostu	Yeterince değil	Daha kullanıcı dostu
Kullanım Kolaylığı	İyi (kolay değil)	Oldukça iyi (yeterince kolay)
Lisans	GPLv3 / Ticari	LGPLv3
Performans	Oldukça iyi	Çok zayıf

Bu çalışmada, bu kütüphanelere ait işlevlerin bir arada kullanılabilmesi bir program geliştirilmiştir. Dikkat çeken en önemli noktalardan biri, Emgu CV'nin hesaplamasal olarak oldukça verimli olduğu ve özellikle gerçek-zamanlı uygulamalarda uygun olduğu yönündedir. Öte yandan, OpenCV bilgisi gerektirmeksizin doğrudan ve kolayca kullanılabilmesi ise AForge.NET için en belirgin avantajlardan biridir. Bununla birlikte, resim kutucukları ile son derece esnek kullanım için Emgu CV tercih edilebilir. Buradaki ortak nokta ise mevcut kod örneklerinin ve arayüz tasarımlarının büyük ölçüde geliştirme zamanını düşüreceğidir. Bu örnekler gerektiğinde düzenlenebilir, fakat düzenlenecek/silinecek koddaki beklenmeyen davranış dikkate alınmalıdır.

Görsel arayüz için Emgu CV'ye ait arayüz bileşenleri ile form tasarımı yapılırken ve çeşitli Gİ tekniklerinden yararlanırken, AForge.NET ile birtakım morfolojik ve dönüşümsel işlemler yapılıyor olabilir. C# geliştiricileri, bu çalışmada olduğu gibi, herhangi bir Gİ çalışması için her iki kütüphaneyi de kullanarak yazılım geliştirebilirler. Bu kapsamda, filtreler, kenar haritaları, dönüşüm ve bölütleme gibi temel Gİ teknikleri; tespit, izleme ve eşleme amacıyla gerçekleştirilen öznitelik analizi veya yapay öğrenme, kümeleme ve sınıflandırma işlemleri uygulanabilir.

Görüntüleme, tanıma, izleme, algılama, teşhis, bir fabrikadaki ürünün niteliğinin ölçümü ve araçlar için park destek sistemi gibi amaçlarla geliştirilen tüm Gİ uygulamalarında, problemlere yönelik geliştirilen algoritmalarla ziyade; görüntülerin yakalanması, sunumu ve görüntüleme ön işleme uygulama (görüntü iyileştirme) son derece önemlidir. Çünkü problemlerin doğru ve verimli bir biçimde çözümü, görüntülerden çıkarılacak ön bilgilerin niteliğine ve gürbüzlüğüne bağlıdır. Ayrıca bu durum başarıyı doğrudan etkilemektedir ve pek çok Gİ tekniğinin farklı sıralamalarda ve birleşimlerde uygulanması gerekebilir. Bu nedenle Gİ araçlarına gerek duyulur. Bu kapsamda, sözü geçen uygulamaları gerçeklemede, Gİ aracı olarak bir program yazılmıştır. Bu çalışmanın, Visual Studio geliştirme platformunda (gürbüzdür ve ücretsiz olarak sağlanabilir) C# ile Gİ geliştiricileri için Emgu CV ve AForge.NET kullanımına örnek niteliğinde olabileceği düşünülmektedir. Karşılaştırma ölçütlerine bakıldığında, kapsamlı uygulamalar geliştirmede her iki kütüphanenin de kullanımının gerekebileceği aşikârdır.

5. KAYNAKLAR (REFERENCES)

- [1]. Güney, M. ve Arıca, N., (2009). Desen Tabanlı İlgili Bölgesi Tespiti, *Journal of Naval Science and Engineering*, 5(1), 94-106.

- [2]. Özcan, M. O., Sert, E., Taşkın, D. ve Taşkın, C., (2013). OpenCV ile Stereo Görüntülerden Derinlik Kestirimi, *Akademik Bilişim 2013*.
- [3]. *OpenCV*, <http://opencv.org/> [Erişim Tarihi: 19 Temmuz 2015].
- [4]. Doğan, T., Sert, E. ve Taşkın, D., (2013). Araç Destek Sistemleri İçin Kuş Bakışı Görüntü Dönüşümü, *Akademik Bilişim 2013*.
- [5]. Çomak, B., Beycioğlu, A., Başyigit, C. ve Kılınçarslan, Ş., (2011). Beton Teknolojisinde Görüntü İşleme Tekniklerinin Kullanımı, *6th International Advanced Technologies Symposium (IATS'11)*, 16-18 May, Elazığ, Turkey.
- [6]. Aksu, S., Demirel, H. ve Görgünoglu, S., (2013). Bulut Bilişim Teknolojisi ile Bir Görüntü İşleme Uygulaması, *Akademik Bilişim 2013*.
- [7]. Varlık, A. ve Çorumluoğlu, Ö., (2011). Dijital Fotogrametri Teknikleri İle Kişi Tanıma, *Harita Teknolojileri Elektronik Dergisi*, 3(2), 1-24.
- [8]. *Emgu CV: OpenCV in .NET (C#, VB, C++ and more)*, http://www.emgu.com/wiki/index.php/Main_Page [Erişim Tarihi: 19 Temmuz 2015].
- [9]. *AForge.NET :: Computer Vision, Artificial Intelligence, Robotics*, <http://www.aforge.net.com/> [Erişim Tarihi: 29 Temmuz 2015].
- [10]. Toth, S. *Open Source .NET Libraries for Image Processing, Recognition and Computer Vision*, <http://www.stefantoth.com/content/papers/> [Erişim Tarihi: 28 Temmuz 2015].
- [11]. Grycuk, R., Gabryel, M., Korytkowski, M., Scherer, R., and Voloshynovskiy, S., (2014). From Single Image to List of Objects Based on Edge and Blob Detection, *Lecture Notes in Computer Science*, 8468, 605-615, DOI: 10.1007/978-3-319-07176-3_53, <http://arxiv.org/pdf/1504.06867.pdf>.
- [12]. Liao, H. C., Yang, C. H., Huang, H. W., and Shin, J., (2015). Consistent Labeling Approach for a PTZ Camera Based on Template Cache and Least Recently Used Replacement Strategy, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(2), 361-374, <http://dx.doi.org/10.14257/ijcip.2015.8.2.34>.
- [13]. *.NET Framework wrapper for OpenCV*, <https://github.com/shimat/opencvsharp> [Erişim Tarihi: 29 Temmuz 2015].
- [14]. *.NET Framework Wrapper for Intel's OpenCV Package*, <https://code.google.com/p/opencvdotnet/> [Erişim Tarihi: 31 Temmuz 2015].
- [15]. Gade, R., and Moeslund, T. B., (2014). Thermal Tracking of Sports Players, *Sensors 2014*, 14, 13679-13691, DOI:10.3390/s140813679.
- [16]. Philip, A.A. and Omotosho, M.M., (2013). Image Processing Techniques for Denoising, Object Identification and Feature Extraction, *Proceedings of the World Congress on Engineering*, III, WCE 2013, July 3-5, London, U.K.
- [17]. Tayade, M. R., Jeyakumar, A., Kore, A. B., and Galshetwar, V. M., (2014). Real Time Eye State Monitoring System for Driver Drowsiness Detection, *International Journal of Emerging Technology and Advanced Engineering*, www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, 4(6), June).
- [18]. Meena, A., and Raja, K., (2014). K-Means Segmentation of Alzheimer's Disease In Pet Scan Datasets – An Implementation, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 117, 168-172, <http://arxiv.org/ftp/arxiv/papers/1302/1302.7082.pdf>.
- [19]. Rygał, J., Romanowski, J., Scherer, R., and Ferdowsi, S., (2014). Novel Algorithm for Translation from Image Content to Semantic Form, *Lecture Notes in Computer Science*, 8467, 783-792, DOI: 10.1007/978-3-319-07173-2_67, http://sip.unige.ch/articles/2014/JointPaper_ICAISC2014.pdf.
- [20]. Köksal, T. ve Şaykol, E., (2013). 360 Derece Hareket Algılayan Bir Çevre Güvenlik Sistemi Uygulaması, *Beykent University Journal of Science and Engineering*, 6(1), 57-76.
- Ahmed, S. M. H., Alexander, T. C., and Anagnostopoulos, G. C. Real-time, Static and Dynamic Hand Gesture Recognition for Human-Computer Interaction, http://www.amalthea-reu.org/pubs/amalthea_tr_2008_01.pdf [Erişim Tarihi: 27 Kasım 2015].