Test Kâhini Olarak Görüntü Karşılaştırma Algoritmalarının Değerlendirilmesi

Ömer Faruk Erdil[1], İrfan Can[1], Hasan Sözer[2]
[1] Vestel, Manisa, Türkiye
{faruk.erdil, irfan.can}@vestel.com.tr
[2] Özyeğin Üniversitesi, İstanbul, Türkiye
hasan.sozer@ozyegin.edu.tr

Özet. Televizyon gibi yoğun yazılım içeren gömülü sistemlerin kara kutu testleri, grafik kullanıcı arayüzleri (GKA) aracılığıyla gerçekleştirilmektedir. Bu testlerin otomasyonu kapsamında bir dizi kullanıcı işlemi dışarıdan tetiklenmektedir. Bu sırada, doğru ve yanlış sistem davranışı arasında ayrım yapan ve böylece testlerin geçip geçmediğine karar veren otomatik bir test kâhinine ihtiyaç duyulmaktadır. Bu amaçla yaygın olarak görüntü karşılaştırma araçları kullanılmaktadır. Bu araçlar, gözlenen GKA ile daha önceden kaydedilmiş bir referans GKA ekran görüntüsünü karşılaştırmaktadır. Bu çalışmada, 9 farklı görüntü karşılaştırma aracı bir endüstriyel vaka çalışması ile değerlendirildi. Bir televizyon sisteminin gerçek test çalışmalarından 1000 çift referans ve anlık GKA görüntüsü toplandı ve bu görüntüler başarılı/başarısız test olarak etiketlendirildi. Ayrıca, toplanan veri kümesi görüntülerde meydana gelen piksel kayması, renk tonu/doygunluk farklılığı ve resim gövdesinde esneme (büyüme, küçülme, genişleme, daralma) gibi çeşitli etkilere göre sınıflandırıldı. Ardından, bu veri kümesi ile karşılaştırılan araçlar, doğruluk ve performans açısından değerlendirildi. Araçların parametre değerlerine ve karşılaştırılan görüntülerin tâbi oldukları etkilere göre farklı sonuçlar verdiği görülmüştür. Hazırlanan veri kümesi için en iyi sonuçları veren araç ve bu aracın parametre değerleri tespit edilmiştir.

Anahtar kelimeler: Görüntü Karşılaştırması, Görüntü Karşılaştırma Algoritmaları, Test Kâhini, Test Otomasyonu, Endüstriyel Vaka Çalışması.

Evaluation of Image Comparison Algorithms as Test Oracles

Ömer Faruk Erdil[1], İrfan Can[1], Hasan Sözer[2]
[1] Vestel, Manisa, Turkey
{faruk.erdil, irfan.can}@vestel.com.tr
[2] Ozyegin University, Istanbul, Turkey
hasan.sozer@ozyegin.edu.tr

Abstract. Black box testing of software intensive embedded systems such as TVs is performed via their graphical user interfaces (GUI). A series of user events are triggered for automating these tests. In the meantime, there is a need for a test oracle, which decides if tests pass or fail by differentiating between correct and incorrect system behavior. Image comparison tools are commonly used for this purpose. These tools compare the observed GUI screen during tests with respect to a previously recorded snapshot of a reference GUI screen. In this work, we evaluated 9 image comparison tools with an industrial case study. We collected 1000 pairs of reference and runtime GUI images during test activities performed on a real TV system and we labeled these image pairs as passed and failed tests. In addition, we categorized the data set according to various effects observed on images such as pixel shifting, color saturation and scaling. Then, this data set is used for comparing tools in terms of accuracy and performance. We observed that results are dependent on tool parameters and various image effects that take place. We identified the best tool and its parameter set for the collected data set.

Keywords: Image Comparison, Image Comparison Algorithms, Test Oracles, Test Automation, Industrial Case Study.

1 Giriş

Gömülü sistemlerde yazılım yoğunluğu ve karmaşıklığı sürekli artmaktadır. Örneğin, geçmişte elektromekanik cihazlar olan televizyon (TV) sistemleri, günümüzde 20 milyon satırdan fazla kaynak kodu içermektedir. Bu eğilim, ürünlere yeni işlevler ve özellikler eklenerek devam etmektedir. Sonuç olarak, bu sistemlerin güvenilirliğini sarsabilecek yeni tehditler belirmektedir. Sistemi eksiksiz test edebilmek ve sistemdeki tüm hataları belirleyebilmek için çok fazla kaynak gerekmektedir [1,10].

Çoğu sistemin kara kutu testleri grafiksel kullanıcı arayüzleri (GKA) aracılığıyla gerçekleştirilmektedir. Bu durum TV gibi gömülü sistemler için de geçerlidir. Bu sistemlerin testinde maliyeti azaltmak amacıyla test otomasyonu sıklıkla tercih edilmektedir [2,14]. Bu kapsamda, olası kullanıcı senaryolarını taklit etmek için önceden belirlenmiş sırada bir dizi kullanıcı işlemi otomatik olarak tetiklenmektedir. Bununla birlikte, doğru ve yanlış sistem davranışını ayırt edebilen ve testin geçip geçmediği-

ne/kaldığına karar veren otomatik test kâhinine (test oracle [4]) ihtiyaç duyulmaktadır. Bu amaçla yaygın olarak görüntü karşılaştırma araçları kullanılmaktadır. Bu araçlar, gözlenen GKA ile daha önceden kaydedilmiş bir referans GKA ekran görüntüsünü karşılaştırmaktadırlar.

Bu bildiride bir dizi alternatif görüntü karşılaştırma tekniği/algoritması ve bunları gerçekleyen 9 farklı araç değerlendirilmektedir. Değerlendirilen araçlar *Perceptual Image Diff (PDIFF)* [13], *ImageMagick (IM)* [5] ve *DSSIM* [3] gibi literatürde adı geçen araçlardan; Vestel için ve/veya Vestel'de geliştirilen/özelleştirilen *Black Transparent (BT)*, *Advanced Fuzzy (AF)* gibi şirkete özel araçlar ile *PSNR* ve *SSIM* [15] gibi kaynak kodu kapalı kütüphanelerden; *Python Imaging Library (PIL)* [12] ve *Python OpenCV Library (CV2)* [11] gibi farklı parametreler alan, geliştirilebilir açık kaynak kodlu kütüphanelerden oluşmaktadır. AF, PSNR, SSIM gibi literatürde adı geçen algoritmalar/araçlar Vestel'de özelleştirilmiştir. PIL ve CV2 gibi literatürde adı geçen kütüphaneler farklı özellikler/parametreler eklenerek Vestel'e özgü görüntü karşılaştırma araçları haline getirilmiştir.

Bir TV sisteminin gerçek zamanlı çalışmasından elde edilmiş anlık resimler ile referans resimlerden oluşan 1000 çift resimlik bir veri kümesi hazırlanmıştır. Bu kümedeki görüntüler test sonuçlarına göre başarılı ya da başarısız olarak etiketlendirilmiştir. Görüntüler, yazılımdan ya da donanımdan kaynaklı etkilere göre piksel kayması, renk tonu/doygunluğu farklılığı ve esneme (resmin gövdesinin genişlemesi, daralması, büyümesi, küçülmesi, vb.) etkilerine tâbi olan resimler olarak sınıflandırılmıştır. Daha sonra, bu veri kümesi görüntü karşılaştırma algoritmalarının/araçlarının doğruluğunu ve performansını karşılaştırmak ve değerlendirmek amacıyla kullanılmıştır.

Genel olarak doğruluk ve performans açısından CV2 diğerler araçlardan üstün olarak görülse de, başarımın parametre değerlerine bağlı olarak ve sınıflandırılmış farklı alt kümeler (piksel kayması, renk doygunluğu, esneme) için önemli oranda değiştiği gözlemlenmiştir. Dolayısıyla, en doğru sonuçlara ulaşmak için test sırasında elde edilecek/edilen görüntünün türüne ve özelliklerine göre kullanılacak aracın parametrelerinin seçilmesi ya da kombine edilmesi gerekmektedir.

Bildirinin organizasyonu: Bir sonraki bölümde, değerlendirmede kullanılan araçlar hakkında genel bilgiler sunulmaktadır. 3. Bölümde vaka çalışması için izlenen yöntem anlatılmaktadır. Bu kısımda, Vestel test ortamı ve TV projeleri hakkında bilgiler verilmekte, hedef ve araştırma soruları açıklanmakta, veri kümesi ve algoritmaların/araçların sınanma şekli hakkında bilgi verilmektedir. 4. Bölümde deney sonuçları paylaşılarak değerlendirilmektedir. 5. Bölümde literatürdeki ilgili çalışmalar özetlenmektedir. Son olarak, 6. Bölümde temel çıkarımlar özetlenerek ileriye dönük çalışmalar listelenmektedir.

2 Deneysel Kurulum

Bu bölümde, endüstriyel vaka incelemesi ve görüntü karşılaştırma araçlarının/algoritmalarının değerlendirilmesi için kurulan deney ortamı anlatılmaktadır.

2.1 Durum Açıklaması

Vaka incelemesi, Vestel'de TV seti geliştirme projesi bağlamında gerçekleştirilmektedir. Geliştirilen ürünler düzenli regresyon testlerine tâbidir. Bu testler, Vestel tarafından geliştirilen test otomasyon araçları ile otomatik hale getirilmektedir. Vestel AR-GE Tasarım Doğrulama ve Test Grubu'nda, TV setlerinin işlevselliği görüntü karşılaştırmasıyla (%65), video analiziyle (%5), ses karşılaştırmasıyla (%2), ürün çıktılarını okuma ve anahtar kelimeler bulmayla (%25), OCR (Optik Karakter Tanıma, %1) ile ve diğer kontrollerle (%2) test edilmektedir. Bu karşılaştırmalar sırasında, referans resim dosyaları (%99) ile referans ses ve çıktı dosyaları (%1) kullanılmaktadır. Bu nedenle, çoğu durumda test senaryolarını değerlendirmek için referans görüntülerle uğraşılmaktadır. Bu görüntüler, projeye ait özelliklerden, kullanıcı arayüzünden, panel yapılandırmasından ötürü farklılıklar gösterebilmektedir.

Görüntü karşılaştırması için kullanılan tüm referans resimler, testler sırasında, Vestel test otomasyon araçları ile yönetilmektedir. Bu araçlar sayesinde, önceden tanımlanmış python betikleriyle (script) uzaktan kumanda komutları TV'ye gönderilmekte, önceden belirlenmiş yürütme noktalarında anlık görüntüler alınmakta ve bu görüntüler referans resimlerle karşılaştırılmaktadır.

2.2 Karşılaştırılan Araçlar

Vaka çalışması kapsamında değerlendirilen araçlar **Tablo 1**'de özellikleri ile listelenmiştir.

Tablo 1. Değerlendirme için Kullanılan Araçlar

Kısaltma İsmi	Kaynak Kodu	Platform	Parametre
PSNR	Kapalı	Windows	Yok
SSIM	Kapalı	Windows	Yok
AF	Kapalı	Windows, Mac OS, Linux	Yok
BT	Kapalı	Windows, Mac OS, Linux	Yok
DSSIM	Kapalı	Windows, Mac OS, Linux	Yok
PDIFF	Kapalı	Windows, Mac OS, Linux	Yok
IM	Kapalı	Windows, Mac OS, Linux	Bulandırma
PIL	Python	Windows, Mac OS, Linux	Gri mod, Tolerans
CV2	Python	Windows, Mac OS, Linux	Eşik değeri, Blok boyutu, Kenar silme, Küçültme

2.3 Hedef ve Araştırma Soruları

Bu çalışmadaki amacımız, araçları/algoritmaları 'doğruluk' ve 'performans' ölçütlerine göre karşılaştırmaktır. Görüntü karşılaştırma araçlarının doğruluğunu etkileyebilecek birçok görüntü etkisi bulunmaktadır. Ayrıca, karşılaştırılacak görüntüler piksel kayması, doygunluk ve esneme gibi farklı etkilere maruz kaldığında görüntü karşılaştırma araçlarının/algoritmalarının nasıl performans göstereceği bilinmek istenmektedir. Dolayısıyla, iki araştırma sorusu tanımlanmaktadır:

1. Her araç için görüntü karşılaştırma sonuçları ne kadar doğrudur?
2. Araçların görüntü karşılaştırma performansları nasıldır?

2.4 Veri Kümesi Koleksiyonu

Test sonuçlarından elde edilmiş bir dizi referans ve anlık resim kullanılarak bir veri seti hazırlanmıştır. Bu setteki görüntüler test sonuçlarına göre başarılı ya da başarısız olarak etiketlendirilmiştir. Bu veri setinde 42 çift piksel kaymasına, 143 çift renk tonu/doygunluğu etkisine, 359 çift esneme etkisine maruz kalmış başarılı resim ve 456 çift başarısız resim bulunmaktadır. Bu resim seti, test ortamında python betiği (script) yardımıyla 9 farklı araçla/algoritmayla ve değişik parametreleriyle birlikte karşılaştırılmıştır (bkz. **Tablo 1**).

2.5 Araçlar, Metrikler ve Değişkenler

Değerlendirme metrikleri açısından, iki kıstas göz önüne alınmaktadır: performans ve doğruluk. Performans metriği, iki görüntünün karşılaştırılması için geçen ortalama (saniye cinsinden) süre olarak ölçülmüştür. Doğruluğu değerlendirmek için kullanılan ölçütler aşağıda listelenmiştir:

- **TP**: Karşılaştırma aracının başarısız test çalışmasında iki resim (referans ve anlık resim) arasında eşleşmezlik fark ettiği durumların sayısı
- **TN**: Karşılaştırma aracının başarılı test çalışmasında iki resim (referans ve anlık resim) arasında hiçbir fark bulamadığı durumların sayısı
- **FP**: Karşılaştırma aracının başarılı test çalışmasında iki resim (referans ve anlık resim) arasında eşleşmezlik fark ettiği durumların sayısı
- **FN**: Karşılaştırma aracının başarısız test çalışmasında iki resim (referans ve anlık resim) arasında hiçbir fark bulamadığı durumların sayısı

Bu ölçütler aşağıdaki gibi doğruluk ölçütünü hesaplamak için kullanılmıştır:

$$\textbf{Doğruluk} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

'ImageMagick' algoritmasında, 'bulandırma' parametresi '0 – 100' aralığında herhangi bir değer alacak şekilde kullanılabilmektedir. Bu çalışmada 6 farklı 'bulandırma' değeri (0, 5, 10, 15, 20, 25) kullanılarak 6 farklı karşılaştırma yapılandırması (konfigürasyonu) elde edilmiştir.

'PIL' algoritmasında, 'gri mod' parametresi 'etkin (1)' ve 'etkisiz (0)' olacak ve 'tolerans' parametresi '0 - 255' aralığında herhangi bir değer alacak şekilde kullanılabilmektedir. Bu çalışmada 4 farklı tolerans değeri (0, 8, 16, 32), 2 farklı 'gri mod' değeriyle kullanılarak 8 farklı karşılaştırma yapılandırması (konfigürasyonu) elde edilmiştir.

'CV2' aracında/algoritmasında, 'kenar silme' parametresi 'etkin (1)' ve 'etkisiz (0)' olacak şekilde, 'blok boyutu' parametresi '5 – 1920' aralığında herhangi bir değer alacak şekilde, 'eşik değeri' parametresi '0 – 255' aralığında herhangi bir değer alacak şekilde ve 'küçültme' parametresi '0.01 – 1.0' aralığında herhangi bir değer alacak şekilde kullanılabilmektedir. Bu çalışmada 15 farklı 'eşik değeri' (15, 31, 47, ... 223), 2 farklı 'kenar silme', 4 farklı 'küçültme' (1.0, 0.5, 0.25, 0.125) ve 21 farklı 'blok boyutu' (5, 6, ... 20, 24, … 120, … 1920) değeri kullanılarak yaklaşık 2500 karşılaştırma yapılandırması (konfigürasyonu) elde edilmiştir. Bu parametreler bazı değer kombinasyonlarıyla gruplandırılmıştır (bkz. **Tablo 2**).

3 Değerlendirme

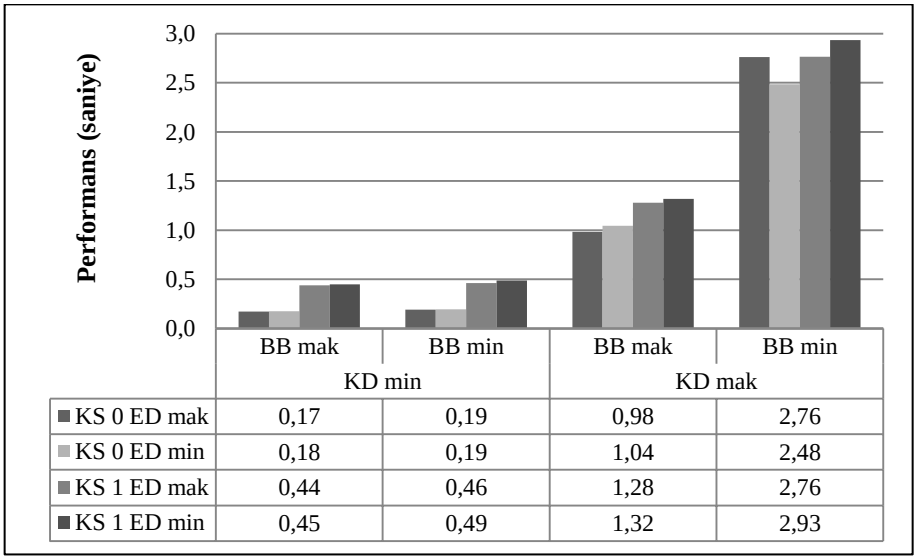
Daha önce bahsedilen 9 görüntü karşılaştırma aracı, piksel kaymasına, renk doygunluğuna ve esnemeye maruz kalmış başarılı resim setlerinde ve başarısız resim setinde kullanılmıştır. 'IM', 'PIL' ve 'CV2' araçları farklı parametrelerle çalıştırılarak daha fazla karşılaştırma çeşitliliği elde edilmiştir. 'IM' ve 'PIL' araçlarında en doğru sonucu veren parametre yapılandırmaları kullanılmıştır. 'CV2' aracında her bir resim seti (Başarısız, Başarılı, Piksel Kayması, Renk Doygunluğu, Esneme) için **en iyi sonucu veren** birer örnek ve genelde **en doğru sonucu veren** toplamda 6 farklı parametre yapılandırması kullanılmıştır (bkz. **Tablo 2**). Bütün araçların değerlendirilmesinde aynı '**limit** (65)' değeri kullanılmaktadır. Bu değere göre test sonuçlarında eşleştirmenin başarılı veya başarısız olduğuna karar verilmektedir.

Tablo 2. 'CV2' Yapılandırması (Konfigürasyonu)

Yapılandırma	Kenar silme	Eşik değeri	Blok boyutu	Küçültme
CV2-y1	1	79	20	0,125
CV2-y2	0	223	20	0,125
CV2-y3	0	159	24	0,125
CV2-y4	1	79	240	0,25
CV2-y5	0	63	60	0,125
CV2-y6	1	79	5	0,5

Not: CV2-y1: En doğru sonucu veren parametre yapılandırması, CV2-y2: Piksel kayması seti için en iyi sonucu veren yapılandırmalardan biri, CV2-y3: Renk doygunluğu seti için en iyi sonucu veren yapılandırmalardan biri, CV2-y4: Esneme seti için en iyi sonucu veren yapılandırmalardan biri, CV2-y5: Başarılı set için en iyi sonucu veren yapılandırmalardan biri, CV2-y6: Başarısız set için en iyi sonucu veren yapılandırmalardan biri olduğu gözlemlenmiştir.

CV2'nin daha önce de değinildiği gibi 2500 civarında farklı parametre değeri varyasyonlarıyla test edilmiş ve bu varyasyonların hepsinin ortalama test süreleri incelenmiştir. CV2 aracının performansına en çoktan en aza doğru sırasıyla 'küçültme' değeri, 'blok boyutu', 'kenar silme' ve 'eşik değeri'nin etki ettiği gözlenmektedir. 'Kenar silme' parametresinin '0', 'eşik değeri'nin '255', 'küçültme' değerinin '0.125' ve 'blok boyutu'nun '1920' seçilmesi durumunda '0.17' saniye ile en hızlı CV2 yapılandırması elde edilmiştir. En yavaş CV2 yapılandırması da 'kenar silme' parametresinin '1', 'eşik değeri'nin '0', 'küçültme' değerinin '1.0' ve 'blok boyutu'nun '5' seçilmesiyle elde edilmiştir ve test süresi **Şekil 1**'de görüldüğü gibi '2.93' saniye olarak ölçülmüştür.



	KD min		KD mak	
	BB mak	BB min	BB mak	BB min
KS 0 ED mak	0,17	0,19	0,98	2,76
KS 0 ED min	0,18	0,19	1,04	2,48
KS 1 ED mak	0,44	0,46	1,28	2,76
KS 1 ED min	0,45	0,49	1,32	2,93

KD min-mak: Küçültme değeri (0.125 – 1.0),
BB min-mak: Blok boyutu (5 - 1920),
ED min-mak: Eşik değeri (0 - 255),
KS 0-1: Kenar silme (Etkin değil - Etkin)

Şekil 1. Minimum/Maksimum Değerlerde 'CV2' Performansı (saniye)

'IM'de en doğru sonucu veren parametre yapılandırması 'bulandırma' parametresinin '5' olduğu ve 'PIL'de ise 'gri mod' parametresinin 'etkin (1)' ve 'tolerans' parametresinin '8' olduğu durumdur.

Tüm görüntü setleri ve algoritmalar **Tablo 3**'te tartışılmaktadır. **Tablo 3**'e göre en yavaş 3 araç sırasıyla 'PDIFF', 'DSSIM' ve 'BT' araçları olarak göze çarpmaktadır. En hızlı araçlar olarak 'CV2' aracının 'küçültme' parametresinin en düşük olduğu (0.125) yapılandırmalar (1, 2, 3, 5) olarak göze çarpmaktadır. 'CV2' aracı dışında en hızlı çalışan 2 araç sırasıyla 'PSNR' ve 'IM' araçlarıdır.

Tablo 3. Değerlendirme için Kullanılan Araçlar ve Yapılandırmalar.

	Piksel Kayması Seti		Renk Doy-gunluğu Seti		Esneme Seti		Başarılı Set		Başarısız Set			
Araç	**TN**	**FP**	**TN**	**FP**	**TN**	**FP**	**TN**	**FP**	**TP**	**FN**	**Doğruluk**	**Süre (s)**
PSNR	29	13	115	28	0	359	144	400	395	61	0,54	0,78
SSIM	42	0	137	6	354	5	533	11	170	286	0,70	1,55
DSSIM	42	0	143	0	359	0	544	0	0	456	0,54	4,89
PDIFF	42	0	109	34	350	9	501	43	173	283	0,67	7,68
AF	42	0	143	0	359	0	544	0	24	432	0,57	1,62
BT	42	0	143	0	359	0	544	0	7	449	0,55	3,65
IM	42	0	120	23	348	11	510	34	203	253	0,71	1,18
PIL	42	0	134	9	349	10	525	19	196	260	0,72	2,50
CV2-y1	29	13	109	34	316	43	454	90	340	116	0,79	0,39
CV2-y2	42	0	119	24	355	4	516	28	94	362	0,61	0,16
CV2-y3	42	0	143	0	319	40	504	40	137	319	0,64	0,16
CV2-y4	33	9	143	0	359	0	535	9	101	355	0,64	0,43
CV2-y5	42	0	143	0	359	0	544	0	103	353	0,65	0,15
CV2-y6	1	41	13	130	1	358	15	529	456	0	0,47	1,11

3.1 Başarısız Görüntü Grubu

Bu grup, 456 çift resimden oluşmaktadır ve gerçek karşılaştırma sonucunun başarısız olduğu bilinmektedir. Ama çeşitli sebeplerden ötürü otomatik karşılaştırmalarda fark görülememektedir (FN). Sonuçlara göre, 'CV2' aracı haricindeki 8 araç içinde %86'lık başarı oranıyla 'PSNR' aracı göze çarpmaktadır. Ancak başarısız görüntü grubunda 'hata kaçırmak' kabul edilemez bir durumdur. Çünkü kaçırılan her hata, ciddi bir risk demektir. Bu durumda 'CV2' aracı devreye girmektedir ve 'CV2–y6' yapılandırması düşük 'blok boyutu' sayesinde hiç hata kaçırmayıp %100 başarıyla bu grubu tamamlamaktadır.

3.2 Piksel Kayması Görüntü Grubu

Bu grup, 42 çift resimden oluşmaktadır ve gerçek karşılaştırma sonucunun başarılı olduğu bilinmektedir. Ama anlık görüntülerde referanslarına göre piksel kaymaları olduğu için otomatik karşılaştırmalarda başarısız sonuç elde edilebilmektedir (FP). Sonuçlara göre, bu görüntü grubu için araçlar genel olarak başarılı bir performans ortaya koyarken 'CV2' aracının 'kenar silme' parametresinin '1 (etkin)' olduğu yapılandırmalar (1, 4, 6) ile 'PSNR' aracı en başarısız araçlar olarak göze çarpmaktadır.

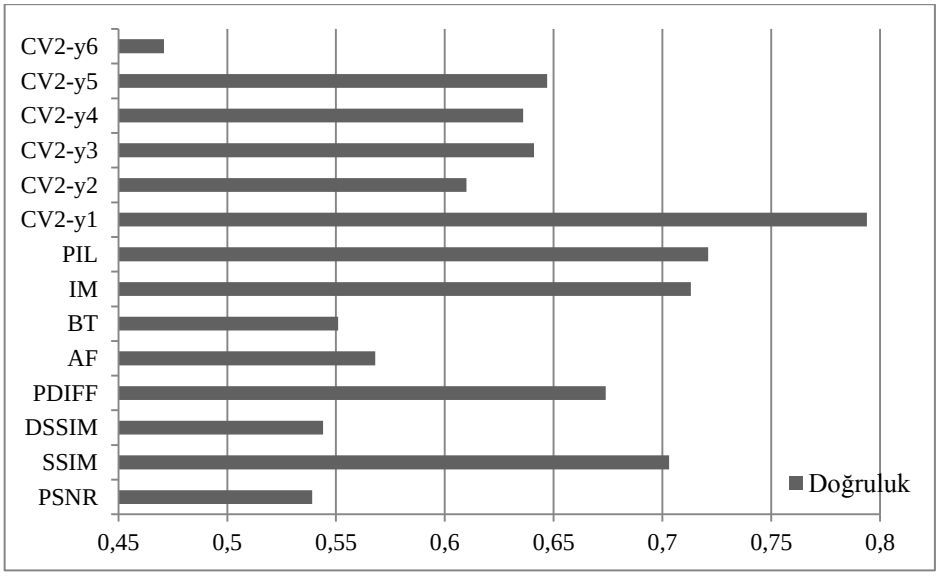
3.3 Renk Tonu/Doygunluğu Görüntü Grubu

Bu grup, 143 çift resimden oluşmaktadır ve gerçek karşılaştırma sonucunun başarılı olduğu bilinmektedir. Ama anlık görüntülerin referanslarına göre farklı renk tonu yoğunluğundan dolayı otomatik karşılaştırmalarda başarısız sonuç elde edilebilmektedir (FP). Sonuçlara göre, bu görüntü grubunun testlerinde 'BT', 'AF', 'DSSIM' ve 'CV2'nin 2, 3 ve 5. yapılandırmalarında %100 başarı gözlenmektedir. Diğer yandan, 'CV2'nin 6. yapılandırması en başarısız araç olarak göze çarpmaktadır (%9).

3.4 Esneme Görüntü Grubu

Bu grup, 359 çift resimden oluşmaktadır ve gerçek karşılaştırma sonucunun başarılı olduğu bilinmektedir. Ama anlık görüntülerin referanslarına göre farklı şekillerde esnemeye maruz kalmasından dolayı otomatik karşılaştırmalarda başarısız sonuç elde edilebilmektedir (FP). Sonuçlara göre, 'CV2'nin 4 ve 5. yapılandırmasında ve 'AF', 'BT' ve 'DSSIM' araçlarında %100 başarı elde edilmektedir. Diğer yandan, en başarısız araçlar olarak 'PSNR' (%0) ve 'CV2'nin 6. yapılandırması (%0.02) görülmektedir.

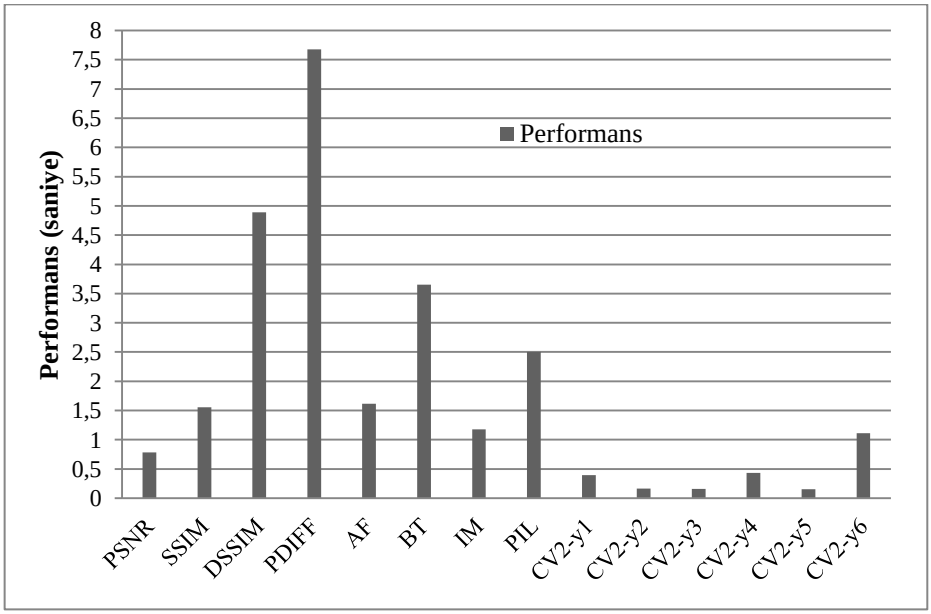
Genel olarak başarılı görüntü grupları incelendiğinde en başarılı sonuçlar %100 ile 'BT', 'AF', 'DSSIM' ve 'CV2-y5' araçlarından elde edilmektedir. 'CV2-y5' aracının diğer üç araçtan daha hızlı olması (0.15 sn.) ve doğruluğunun diğerlerinden daha yüksek olması (0.65) sebebiyle başarılı görüntü grupları için en tercih edilir araç olarak göze çarpmaktadır (bkz. **Şekil 2, Şekil 3**).



Şekil 2. Görüntü Karşılaştırma Araçlarının Doğruluk Grafiği

En doğru sonucu veren aracı bulmak için başarılı görüntü grubunun yanında başarısız görüntü grubu da incelenmeli ve bütün gruplar temel alındığında ortaya çıkan doğruluk ve performans sonuçlarına göre karar verilmelidir.

Başarısız görüntü grubu incelendiğinde en iyi sonuçları sırasıyla 'CV2-y6', 'PSNR' ve 'CV2-y1' araçları vermektedir. Bu üç aracın performansı '1' saniyenin altında olduğundan doğruluk temel alındığında ise en iyi sonucu '**0.79**'la '**CV2-y1**' aracı vermektedir. Başarılı ve başarısız setler bir arada değerlendirildiğinde ise en doğru sonucu veren araç olarak '**CV2-y1**' aracı ön plana çıkmaktadır (bkz. **Şekil 2, Şekil 3**).



Şekil 3. Karşılaştırma Araçlarının Performans (saniye) Grafiği

'CV2-y1' aracı bütün gruplarda en iyi sonucu vermemesine rağmen, genel olarak başarımının diğer araçlardan üstün olduğu görüldü. Bu araçtan sonra en doğru sonucu veren araçlar parametre alabilen 'PIL' ve 'IM' araçlarıdır. Diğer taraftan, en yanlış sonucu veren araç/yapılandırma '0,47' ile 'CV2-y6' aracıdır. Buradan, parametre seçiminin aracın doğru ya da yanlış sonuç üretmesine önemli oranda etkisi olduğu görülmektedir.

Tüm sonuçlar incelendiğinde 'DSSIM' aracının, başarılı görüntü setinde %100 başarılı olmasına rağmen başarısız görüntü setinde hiç hata yakalayamaması dikkat çekmektedir. Bu da 'DSSIM' aracının karşılaştırma sırasında resimler arasında hiçbir fark ayırt edemediğini ve bu yüzden bu aracın tercih edilmemesi gerektiğini göstermektedir.

'PSNR' aracı başarısız görüntü setinde 'CV2' dışındaki araçlara göre çok daha başarılı (%86) olmasına rağmen esneme setinde %100 başarısız olması dikkat çekici

noktalardan biridir. Bu da resimlerin esnemeye maruz kalma ihtimalinin olduğu durumlarda, bu aracın tercih edilmemesi gerektiğini göstermektedir.

Performans açısından genel bir değerlendirme yapıldığında ise 'PDIFF' aracının diğer araçlara göre önemli derecede yavaş olduğu görülmektedir. En hızlı araç 'CV2-y5'ten yaklaşık 50 kat, en başarılı araç 'CV2-y1'den yaklaşık 20 kat daha yavaş sonuç vermektedir. Bu aracı sırasıyla 'DSSIM' ve 'BT' araçları takip etmektedir. Doğruluk açısından da diğer araçlar ile kıyaslandığında vasat bir başarım gösterdiğinden, bu aracın tercih edilmemesi gerektiği görülmektedir.

Özetle, performans ve doğruluk açısından sırasıyla 'CV2-y1', 'PIL', 'IM' ve 'SSIM' araçları tercih edilebilir araçlar olarak öne çıkmaktadır. Diğer yandan, görüntülerin esneme etkisine maruz kalma ihtimalinin düşük olduğu durumlarda 'PSNR' aracı da tercih edilebilir araçlar arasında değerlendirilebilir.

4 İlgili Çalışmalar

Görüntü karşılaştırması üzerine literatürde yapılmış birçok çalışma incelenmiştir ve bu çalışmalarda kullanılan birçok farklı yöntem, teknik ve araç hakkında bilgiler verilmektedir. Ancak bilgimiz dahilinde bu yöntemler endüstriyel bir vaka çalışması kapsamında değerlendirilerek karşılaştırılmamışlardır.

Sıkıştırılmış bilgisayarlı tomografi görüntülerinin, görüntü kalitesinin görsel değerlendirilmesi için görüntü karşılaştırma yöntemleri karşılaştırılmıştır [7]. Araştırmanın amacı, farklı görüntü sıkıştırma algoritmaları ile sıkıştırılan aynı görüntünün, görüntü doğruluğunu karşılaştırarak görüntü sıkıştırma için en iyi yöntemi bulmaktır. Ana konu tıbbi imge karşılaştırmalarına dayansa bile, bu araştırmanın genel sonuçları nedeniyle bu karşılaştırmanın sonucu her alana uygulanabilmektedir. Tıbbi alanda görüntü kalitesi çok önemlidir, çünkü görüntü bozulması doğru tanıyı engelleyebilmekte ve hasta sağlığında büyük sorunlara neden olabilmektedir. Resim doğruluğunun, görüntü karşılaştırma yöntemleriyle kontrol edilmesi gerekmektedir. Makaleye göre, görüntü karşılaştırmaları için üç yöntem kullanılmaktadır; *i)* araya giren boş bir görüntü olmadan alternatif ekran (AWOB), *ii)* araya giren boş bir görüntüye sahip alternatif ekran (AWB) ve *iii)* yan yana görüntü (SSD). Sonuçlar, AWOB yönteminde görüntü farkına duyarlılığın anlamlı düzeyde yüksek olduğunu ve AWB ile SSD arasında çok az fark olduğunu göstermektedir. Araştırma tekniği ve görüntü karşılaştırma yöntemlerinin, bilgisayar uygulamaları yerine insan gözlemcilerini içermesi bu yazıyı farklı kılmaktadır. Bu belge sayesinde görüntü karşılaştırmasında farklı bir perspektif elde edilmektedir.

Akıllı cihazlarda kamera ve görüntü işleme performansı giderek önem kazanmaktadır. Kamera ile bütünleşmiş mobil uygulamaların çoğunda bilgisayar görme teknolojisi kullanılmaktadır. QR barkod algılama ve yüz tanıma yazılımı örnek bir uygulama olarak düşünülebilir. Bu kapsamda çeşitli görüntü işleme algoritmaları Android cihazları üzerinde test edilerek karşılaştırılmıştır [16]. Karşılaştırmada 3 farklı yöntem değerlendirilmiştir: *i)* Ölçek Değiştirme Özelliği Dönüşümü (SIFT), *ii)* Sağlam Özellikleri Hızlandırma (Surf) ve *iii)* Kısa Yönlendirilme (ORB). ORB algoritmasının

performans açısından en ‘hızlı’ olduğu, ancak herhangi bir yöntemi seçmek yerine karma bir yöntemin kullanılmasının daha iyi bir seçenek olduğu görülmüştür.

Önceki bir çalışmada [6], piksel-piksel karşılaştırmaları, Hausdorff mesafesi, uzaklık tabanlı işlevler, görüntü değiştirme algılama algoritmaları, ön işleme yöntemleri, renk tutarlılık vektörleri ve ortak histogramlar gibi farklı yöntemler açıklanarak, bu yöntemleri uygulayan görüntü karşılaştırma araçları değerlendirilmiştir. Değerlendirilen araçlar arasında ImageMagick, perceptual diff, image comparer, image diff, ImageJ, bio7, gimp ve OpenCV araçları bulunmaktadır. İlgili bildiride [6], detaylı raporlanmış bir vaka çalışması kapsamında bu araçların bir kıyaslaması bulunmamakla birlikte, ayrı ayrı bu araçların nasıl kullanılacağına dair ipuçları verilmiştir.

5 Sonuç

TV setlerinin test otomasyonunda, testlerin başarılı ya da başarısız olduğuna karar vermek için TV setlerinin anlık görüntüleri, önceden tanımlanmış referans görüntülerle karşılaştırılmaktadır. Bu çalışmada, 1000 çift resimden oluşan 4 farklı görüntü grubu 9 farklı karşılaştırma aracı/algoritması ile ve parametre alan araçlar çeşitli yapılandırmalarda (konfigürasyonlarda) test edilmiştir, sonuçlar değerlendirilmiştir.

Genel olarak doğruluk ve performans açısından CV2 diğerler araçlardan üstün olduğu görülmüştür. Diğer yandan, karşılaştırma süresinin ve başarımın parametre değerlerine bağlı olarak ve sınıflandırılmış farklı görüntü alt kümeleri (piksel kayması, renk doygunluğu, esneme) için önemli oranda değiştiği görülmüştür. Testler sonucunda her aracın güçlü ve zayıf yanları olduğu, araçların parametre değerlerine ve karşılaştırılan görüntülerin tâbi oldukları etkilere göre farklı sonuçlar verdiği görülmüştür.

Parametre alan araçların belirli bir yapılandırma ile diğerlerinden daha başarılı olduğu, açık kaynaklı araçların geliştirilmeye açık ve performans açısından diğerlerinden daha üstün olduğu görülmüştür. Hazırlanan veri kümesi için, parametre alan araçlara ilişkin en iyi sonuçların alındığı parametre değerleri tespit edilmiştir. Parametre alan araçlarda, hangi parametrenin performansa ne ölçüde etki ettiği belirlenmiştir.

İleride bu çalışma, yeni geliştirilen araçlarla [8,9], farklı testlere özgü gelişmiş veri kümeleriyle ve bu veri kümelerinin maskelenmiş durumlarıyla genişletilebilir. Ayrıca, geliştirilmeye açık araçlara yeni özellikler/parametreler eklenerek en doğru yapılandırma değerleri belirlenebilir.

Kaynaklar

1. Beizer B.: Software Testing Techniques, Van Nostrand Reinhold Co., ed.2 (1990)
2. Berner S., Weber R. and Keller R.K.: Observations and lessons learned from automated testing. In Proceedings of the 27th International Conference on Software Engineering, 571-579 (2005)
3. DSSIM, https://github.com/pornel/dssim, online erişim (2017)
4. Howden, W.E: Theoretical and empirical studies of program testing. IEEE Transactions on Software Engineering, Vol. 4, No. 4, 293-298 (1978)
5. ImageMagick, https://www.imagemagick.org/script/index.php, online erişim (2017)

6. Katukam, R., Sindhoora P.: Image Comparison Methods & Tools: A Review. In Proceedings of the 1st National Conference on Emerging Trends in Information Technology, 35-42 (2015)
7. Kim, B., Lee, H., Kim, K. J., Seo, J., Park, S., Shin, Y.-G., Kim, S. H. and Lee, K. H.: Comparison of three image comparison methods for the visual assessment of the image fidelity of compressed computed tomography images. Med. Phys., 38: 836–844 (2011)
8. Kirac F., Aktemur B. and Sozer H.: VISOR: A fast image processing pipeline with scaling and translation invariance for test oracle automation of visual output systems. Journal of Systems and Software, online https://doi.org/10.1016/j.jss.2017.06.023 (2017)
9. Lin Y. D., Rojas J. F., Chu E. T. H. and Lai Y. C.: On the Accuracy, Efficiency, and Reusability of Automated Test Oracles for Android Devices. IEEE Transactions on Software Engineering, Vol. 40, No. 10, 957-970 (2014)
10. Myers, G.J., Badgett T. and C. Sandler: The Art of Software Testing, John Wiley and Sons Inc., ed. 3 (2012)
11. OpenCV, http://opencv.org/, online erişim (2017)
12. Python Imaging Library (PIL), http://www.pythonware.com/products/pil/, online erişim (2017)
13. Perceptual Image Diff., http://pdiff.sourceforge.net/, online erişim (2017)
14. Rafi D.M., Moses K.R.K., Petersen K. and Mantyla M.V.: Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey. In Proceedings of the 7th International Workshop on Automation of Software Test, 36-42 (2012)
15. SSIM, https://ece.uwaterloo.ca/~z70wang/research/ssim/, online erişim (2017)
16. Yu-Doo K., Jin-Tae P., Il-Young M. and Chang-Heon O.: Performance Analysis of ORB Image Matching Based on Android. International Journal of Software Engineering and Its Applications Vol.8, No.3, 11-20 (2014)