

GERÇEK ZAMANLI GÖMÜLÜ YAZILIMLARDA EKRAN DOĞRULAMASI OTOMASYONU YAKLAŞIMI

Oğuzhan Demir¹
TUSAŞ, Ankara

Efe Gürkan²
TUSAŞ, Ankara

ÖZET

Gerçek zamanlı gömülü yazılımların ekran doğrulamaları hataya oldukça açık bir şekilde, insan gözlemlemesiyle ve uzun zaman alan bir süreç ile yürütülmektedir. En ufak dikkat dağınıklığı bir hatanın gözden kaçmasına sebebiyet vermektedir. Aynı zamanda bu durum, doğrulama yapılırken, doğrulama yapan kişinin bilgisayar başında kalmasını ve zamanını buna ayırmasını zorunlu kılmaktadır. Dolayısıyla, mesai saatleri dışında görsel doğrulama yapmak mümkün olmamaktadır. Bahsedilen problemlerin önüne geçebilmek amacıyla test yazılımlarımıza görsel doğrulamayı otomatikleştirme yetkinliği ekledik. Gömülü sistemimizin DVI çıkışından aldığımız görüntü akışını yakalayan ve Windows ortamına Bitmap olarak aktaran bir ara yüzü ile OpenCV görüntü işleme kütüphanesini kullanarak bu amaca ulaştık. Bunların yanında, kendi Bitmap kıyaslama algoritmalarımızı kullandık. Aynı zamanda testlerin yazım, koşum, raporlama ve bakımını kolaylaştıracak yöntemler entegre ettik. Bu bildiride, yazılımımıza eklediğimiz bu yeteneği gerçekleştirdiğimizi, bu yolda karşılaştığımız sorunlar ve bu otomatikleştirmenin ekibe ve bizlere olan katkısı detaylarıyla açıklanmaktadır.

GİRİŞ

Gömülü sistemlerde özellikle dinamik olan ve anlık değişimlerin sık ve çok önemli bilgiler verdiği ekranların doğrulaması dikkat gerektiren niteliktedir ve hataları gözden kaçırmanın riskler getirebileceği bir alanda yapıyor ise büyük masraflara, hatta bazı projelerde yalnızca mal değil, can kaybına da yol açabilecek sonuçlar doğurabilir. Böyle bir sistemi doğru yapmak keskin çizgileri olan ve hata toleransı düşük olan bir algoritmayla kontrol edildiğinde, bazen insan gözünden kaçabilecek noktaları yakalama şansı verebilir. Bu projenin amacı görsel doğrulamayı bilgisayar kontrolünde yapmak, otomatikleştirmek, dolayısıyla bahsedilen insan hatasından kaynaklanabilecek riskleri minimuma indirmek, görsel doğrulamada zaman kazanabilmek ve de test koşumu yapılırken insan bağımlılığını ortadan kaldırmak olacaktır.

YÖNTEM

Kullanılan Araçlar

Frame Grabber

Ekran görüntüsü yakalayıcılar (frame grabber) herhangi bir kaynaktan video görüntüsü yakalayan küçük taşınabilir aletler veya kartlardır. Bu aletler sayesinde kaydedilen videolar internet yayınında veya herhangi bir yazılımda SDK (Software Development Kit) ara yüzüyle kullanılabilir.

¹ Yazılım Doğrulama Mühendisi, E-posta: oguzhan.demir@tai.com.tr

² Yazılım Doğrulama Mühendisi, E-posta: efe.gurkan@tai.com.tr

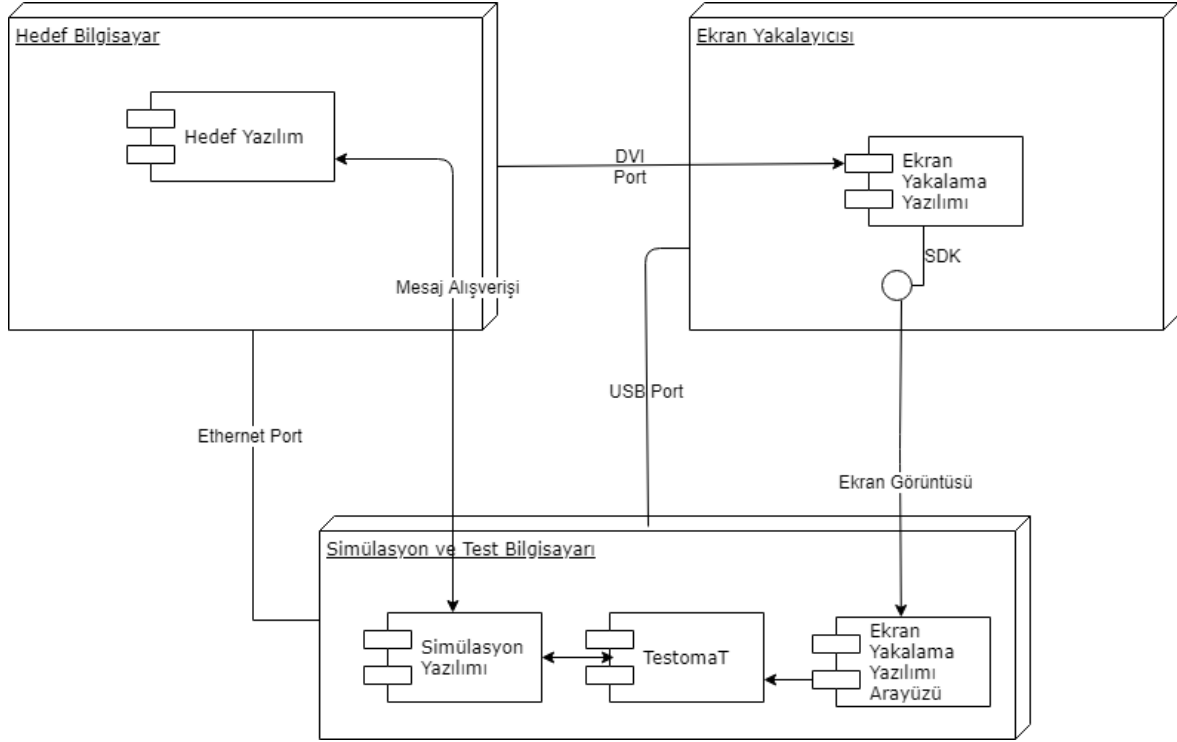
OpenCV

OpenCV (Open Source Computer Vision Library) görüntü işleme üzerine yazılmış açık kaynak kodlu bir kütüphanedir. C ve C++ kullanılarak yazıldığı için birçok görüntü karşılaştırma işleminde optimize ve hızlıdır.

TestomaT

TAI içerisinde geliştirilen ve kullanılan Visual C# üzerinden test senaryosu koşumuna olanak sağlayan, raporlama ve otomatik test koşum gibi özellikleri olan bir programdır. Bahsedilen diğer araçlar son noktada TestomaT'a entegre edilerek kullanılmaktadır.

İş akışı



Şekil 1: Sistem iş akışı

Hedef Yazılım sayesinde Hedef Bilgisayar üzerinden gelen donanımlara ait mesajlar Simülasyon aracına bir Ethernet port ile iletilir. TestomaT Hedef Bilgisayarla Simülasyon Yazılımı aracılığıyla haberleşmektedir. Hedef Bilgisayarda bulunan ekran kartı çıkışı bir Ekran Yakalayıcısı (Frame Grabber)'na bağlıdır. Bu yakalayıcı Hedef Bilgisayar'daki akışı 60 fps/sn. oranına kadar Ekran Yakalama Yazılımına sağlar. Oradan gelen görüntüyü TestomaT alır ve işler.

TestomaT, otomatik ve manuel test koşumuna izin vermektedir. Uygulamalar kısmında manuel bir testin tam anlamıyla nasıl otomatığe dönüştürüldüğü ve bu süreçte alınan kararlar ile iş akışının ayrıntılı açıklamasından bahsedilecektir.

UYGULAMALAR

Yöntemler başlığı altındaki iş akışı takip edilerek TestomaT'a gelen ekran görüntüsünün işlenmesi ile otomatik görsel doğrulama yapılması mümkün olmaktadır. İş akışı şu şekilde gerçekleşir;

- 1- Hedef bilgisayar tarafından doldurulan bir mesaj alanı, normalde bir MFD (multi functional display)'ye gönderilir ve MFD önceden belirlenen şekilde bunu çizip görüntü olarak yansıtır. MFD üzerindeki hard ve soft tuşlar yardımıyla ekranlar arası geçiş yapılabilir veya yeni emirler

verilebilir, bu sayede hedef bilgisayarla iletişim kurulup sistemin durumu değiştirilir. Ekranın da böylece görüntülediği durum değişir.

- 2- MFD'nin bulunmadığı bizim durumumuzda MFD yerine gönderilen mesaj alanları, MFD'yi simüle eden bir Frame Grabber aracı ile DVI kablosu aracılığıyla bilgisayar ekranına yansıtılmaktadır. Hedef bilgisayar görüntü için mesaj alanı göndermesinin yanı sıra normalde hava aracındaki bilgisayara gönderdiği emirleri simülasyon ve test bilgisayarına gönderir. Bu emirlerin doğru gelip gelmediği mesaj alanları kontrol edilerek doğrulanır ve görsel doğrulamaya ihtiyaç duymazlar. Görüntü için gönderilen mesajlar Frame Grabber aracının yazılımı sayesinde yakalanır.
 - a) Yakalanan görüntü akış olduğundan, test edilmek istenen sistem durumu veya koşul test içerisinde yaratıldıktan sonraki süre içerisinde, koşullar değişmeden yine ekran yakalama yazılımı sayesinde bir kare alınır.
 - b) Eğer bu testin ilk koşumu ise ve görsel bir doğrulama ise, TestomaT manuel olarak sistemin durumunun yaratılan koşulla uyup uymadığını (test adımının geçip geçmediğini), testi koşan kullanıcıya sorar. Kullanıcı ekran yakalama yazılımdan akan videodaki görüntüye bakarak test adımının geçip geçmediğine karar verir.
 - c) Test adımı geçmişse kullanıcı adının geçtiğini ekranda beliren butonlardan PASS'e basarak belirtir.
 - d) PASS'e basıldıktan sonra TestomaT yakaladığı kareyi kullanıcının önüne getirir ve kullanıcıya görüntüyü kırpması için seçenek sunar, kullanıcı görüntüde kontrol ettiği kısmı kırpar ve GOLDEN IMAGE almış olur. GOLDEN IMAGE sistemin bulunması gereken durumu temsil eder. Bir sonraki koşulda kullanıcıya sorulmaksızın bu resim kullanılacaktır. Eğer kullanıcı resmi kırparken yanlışlık yaparsa TestomaT'in kırılan resmi gösterdiği noktada resmi tekrar kırp özelliğiyle tekrar resmi görür ve yeniden resmi kırma şansı olur. Resim doğru alınmışsa kullanıcı bunu arayüzden doğrular ve resim kaydedilir.
 - e) FAIL'a basılırsa bu test adımı geçmemiş olur ve sistem istenen durumda olmadığı için TestomaT kullanıcıya görüntü sunmaz, resim kaydedilmez.
 - f) Eğer testin ilk koşumu değilse ve sistemin o anki durumuna ait bir GOLDEN IMAGE varsa, TestomaT otomatik olarak Frame Grabber'ın sağladığı video görüntüsünden bir kare alır, elindeki kareyi kullanıcının kırptığı kısmı hatırlayarak kendisi kırpar ve bu resim ile GOLDEN IMAGE'ı karşılaştırır. Karşılaştırma sonrası kullanıcının belirlediği tolerans sınırları içerisinde bir farklılık yoksa TestomaT bu adımı geçti sayar ve o şekilde raporlar.
 - g) Testin ilk koşumu olmasına rağmen test içinde aynı koşul birden çok kez oluşturuluyorsa, ilk oluşturulduğunda alınan GOLDEN IMAGE ile sonraki seferlerde aynı kontrolü yaptığını hatırlayıp kullanıcı müdahalesine ihtiyaç duymadan otomatik şekilde koşumu yapar.
 - h) Testte değişen bir öge olmadıkça, bir kez koşulduktan sonra tüm görsel doğrulamalar otomatik hale gelir. Eğer yeni bir öge eklenir ya da kullanıcı bir ögenin ismini testte değiştirirse TestomaT bu adımın yeniden doğrulanması için kullanıcıya sorar.

Performans Değerlendirmesi

İlk koşum hız artışı ~%37

Otomatik test koşum hız artışı ~%52

Toplu test koşum hız artışı ~%60

Regresyon test koşumu görsel doğrulama otomasyonundan önce %44 civarında iken, tüm testlerin uyarlanmasıyla sonra %80 ila %95 civarına çıkarılması planlanmaktadır. Gece koşumuna da olanak sağlanmış olacak, test başında kalınmadan tamamen otomatik doğrulama sağlanacaktır.

Eski testlerin yeni sisteme uyarlanması için harcanan zaman ile tüm testlerin koşulup ekran öğelerinin kaydedilmesi aşaması sürecin dezavantajlarından. Görsellerin kapladığı depolama alanı da bir başka dezavantaj olarak gözükmemektedir.

ÇALIŞMA MANTIĞI

Test İçinde Uyulması Gereken Düzen

Testte uyulan ve TestomaT'a o anki test adımının nasıl olması gerektiğini söyleyen anahtar değişkenler, test scripti ile TestomaT arasında iletişimi sağlar. Manuel koşumlarda sözel olarak testi koşan kişinin bu test adımında ne beklediği ekranda bir pencerede belirir. Kullanıcı bir yandan hedef bilgisayardan gelen görüntü akışını kontrol etmelidir, diğer yandan da testi yazan kişinin sözel olarak o adımda ne beklediğini ifade ettiği pencereyi okur ve ona göre o test adımının geçip kalmasına karar verir.

Görsel doğrulamayı otomatikleştirirken testi yazan kişi, testi koşan kişinin anlayabileceği bir dilin yanında TestomaT'a da derdini anlatmalıdır. Bunu sadece 2 anahtar değişken kullanarak başarabilmektedir. Bunlar; IMAGE NAME ile IMAGE STATE'tir.

IMAGE NAME kullanıcının hangi ögeyi doğrulamak istiyorsa o ögeye verdiği ismi ifade eder. Eğer açıklayıcı bir isim tercih ediyorsa iki kelime arası alt çizgiler koyarak testi daha sonra geliştirmek isteyen kişilere de faydası dokunacak uzun isimler kullanabilir. Bu isim içinde kullanılan alt çizgi, düz çizgi gibi kelime ayracı olabilecek karakterler TestomaT tarafından kırılır.

IMAGE STATE kullanıcının hangi öge doğrulanıyorsa o ögenin durumuna verdiği ismi ifade eder. Değişkene verilen isim o ögenin o anki durumunu anlatabilecek açıklıkta olmalıdır. Bu da yine IMAGE NAME gibi uzun olabilmektedir. Örnek vermek gerekirse, ekranın aynı noktasında beliren bir onay kutusu düşünün, ögeye verilen isim IMAGE NAME: 'Onay Kutusu', ögenin durumunu gösteren isim ise IMAGE STATE: 'Donanım kapatma sorusu' olabilirken, testi yazan kişi birden fazla donanımın kapatma sorusunun aynı yerde belireceğinden haberdar ise "donanım kapatma sorusu" çok belirsiz kalacaktır. Hatta farklı donanımlar için soru değişiyorsa test adımının kalmasına yol açacaktır. Bunun yerine daha açıklayıcı olmak için farklı donanımların onay sorusu, farklı test adımları şeklinde yazılmalıdır. IMAGE STATE için ise 'X ögesi kapatılacaktır, devam etmek istiyor musunuz?' gibi bir isimlendirme beklenen durumu detaylıca ifade edebilir. Böylece farklı bir ögenin onay sorusu ekranda belirdiğinde farklı bir IMAGE STATE kullanılacağından, TestomaT da o durumu doğrulayacaktır. Eğer bir öge, aynı IMAGE STATElerle isimlendirilirse, TestomaT bunu ekranın o noktasında aynı durumun görülmesi olarak yorumlayacaktır. Yalnızca IMAGE STATEler farklı ise, TestomaT bunu ekranın o noktasında farklı bir durumun oluştuğu şeklinde yorumlar.

Test yazanın görsel doğrulamayı otomatikleştirirkenki sorumluluğu yalnızca IMAGE NAME ve IMAGE STATE anahtar kelimelerini doğru isimlendirmektir. TestomaT kullanıcıyı geri kalan süreçte yönlendirir.

TestomaT İçinde Çalışma Mantığı

- 1- TestomaT, ekranda önceden belirlenen pivot bölgeyi (hedef ekran üzerinde test durumundan bağımsız aynı durumda kalan, referans alınabilecek bölge) güncel video akışından aldığı resimde bulur ve eğer önceden kaydettiği pivot resimdeki koordinatlardan farklı koordinatlara sahip bir resim elde etmişse ilk olarak akışın elindeki pivot ile önceden kaydedilen pivot arasındaki piksel farkı kadar kaydığını farz eder.
- 2- Pivot resim testin bulunduğu klasör içerisinde yeni bir klasör yaratılarak saklanır, klasör bulunamazsa oluşturulur. GOLDEN IMAGE'lar da burada saklanacaktır.
- 3- Herhangi bir doğrulama süreci başlamadan, kayan piksel değerleri değişkenlerde tutulur.
- 4- TestomaT ilk koşul esnasında eğer koşulan adım geçen bir adımsa kullanıcının hedef ekrandan kırptığı dikdörtgen şeklinde olan bölgeyi, IMAGE NAME_IMAGE STATE-A-B-C-D.png düzeniyle, testin bulunduğu klasörün içine bir klasör daha yaratarak png formatında kaydeder. A,B,C,D burada kırılan resmin, tüm resim üzerindeki koordinatlarını belirtmektedir. Bu resim GOLDEN IMAGE olmaktadır.
- 5- Test içinde testi yazan kişi tarafından sağlanan IMAGE NAME ve IMAGE STATE ile TestomaT, önceden kaydedilen resimler (GOLDEN IMAGE'lar) arasında gezerek bu değişkenleri içeren resmi arar.

- 6- Eğer resim bulunamazsa TestomaT kullanıcıdan manuel doğrulama bekler, kullanıcı hem adımın geçtiğini doğrulamalı, hem de hedef resmi kırpmalıdır.
- 7- Eğer IMAGE NAME eşleşir fakat IMAGE STATE bulunamazsa kullanıcıdan yalnızca adımın geçme doğrulaması beklenir, hedef bilgisayarın o an sağladığı resim diğer eşleşen resimlerdeki koordinatlar ve pivot karşılaştırması sonucu elde edilen piksel kayma değerleri de göz önünde bulundurarak kırpılır.
- 8- Eğer iki anahtar değişkene de sahip olan resim bulunursa, TestomaT bulunan resimden aldığı koordinat bilgisini, pivot karşılaştırması sonucu elde ettiği piksel kayma değerlerini de göz önünde bulundurup kullanarak hedef resmi kırpar. Bu adım tamamen otomatik olan adımdır.
- 9- Kırılan görsel piksel piksel klasörde bulunan resim ile karşılaştırılır. Renk değerleri dahi hesaba katılmalıdır, belli bir tolerans üzerinde gelen farklılıklar test adımının kalmasına yol açar. Bu tolerans kullanıcı tarafından da belirlenebilmektedir.
- 10- Bir kere koşulup tüm GOLDEN IMAGE'lar alındıktan sonra, test koşumu GOLDEN IMAGE almayı gerektirecek yeni bir test adımı teste dâhil olmadıkça artık otomatik olacaktır.

SANAL DEĞİŞKEN İSİMLERİ

A-B-C-D: Kaydedilmiş resimlerin hedef ekrandaki dikdörtgensel koordinat aralığının tam sayı değerleri.

GOLDEN IMAGE: Sistemin bulunması gereken durumu gösteren resim. Bu resmin isimlendirmesi IMAGE_NAME_IMAGE STATE–A-B-C-D.png şeklindedir

IMAGE_NAME: Kaydedilecek görüntünün hedef ekrandaki konumuna kullanıcı tarafından verilmiş isim.

IMAGE_STATE: Kaydedilecek görüntünün hedef ekrandaki durumuna kullanıcı tarafından verilmiş isim.

SONUÇ

Basit araçlar ve açık kaynaklı kütüphane kullanımı ile görsel doğrulamaların otomatikleştirilebildiği gösterilmiştir. Otomatik görsel doğrulamanın pratik faydaları ve verimlilik artışındaki katkıları, bu tekniğin kullanımının daha fazla yaygınlaşmasının gerektiğini göstermektedir. Şu an sadece statik görsel doğrulama, yani anlık resimler üzerinden görsel doğrulama yapılmaktadır. Ayrıca bir ögenin bütün resim içerisinde herhangi bir yerde bulunup bulunmaması ile ilgili senaryolar da doğrulanabilmektedir, ileride dinamik görsel doğrulama, yani hareketli görüntü üzerinden görsel doğrulama da var olan sistem üzerine eklenebilir.