KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GEZGİN ORTAMDA GÖRÜNTÜ DAMGALAMA UYGULAMASI

ALİ UÇAR

KOCAELİ 2014

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

YÜKSEK LİSANS TEZİ

GEZGİN ORTAMDA GÖRÜNTÜ DAMGALAMA UYGULAMASI

ALİ UÇAR

Prof.Dr. Yaşar BECERİKLİ
Danışman, Kocaeli Üniv. .....

Prof.Dr. Hasan OCAK
Jüri Üyesi, Kocaeli Üniv. .....

Prof.Dr. A. Coşkun SÖNMEZ
Jüri Üyesi, İTÜ .....

Tezin Savunulduğu Tarih: 15.07.2014

ÖNSÖZ VE TEŞEKKÜR

Akıllı telefonlar erişilebilir fiyatlar ve keyif verici bir performansa ulaştığı andan itibaren hızla yaygınlaşmıştır. Bu gelişmeleri telekomünikasyon alanındaki gelişmelerin takip etmesiyle birlikte akıllı telefonlar insanların hayatlarında vazgeçilmez bir yer almıştır. Artık pek çok oyunun, web sitesinin ve yazılımın farklı platformlardaki mobil versiyonuna rastlamak mümkündür. Görüntü damgalama bir diğer adıyla gizli veri gömme uygulamaları da bilgisayar veya FPGA gibi özel kartlar üzerinde yıllardır gerçekleştirilen ciddi bir işlem yükü olan uygulamalardır. Bu tez çalışmasında mobil platformda görünmez, dayanıklı bir yöntemle görüntü damgalayan bir yazılım geliştirilmesi amaçlanmıştır.

Bu çalışmanın ortaya çıkmasında fikirleri ve tecrübesi ile yol gösteren, birikimlerini benimle paylaşan, gösterdiği sabır, anlayış, verdiği sürekli destek ve yakınlık için, danışmanım saygıdeğer hocam Prof. Dr. Yaşar BECERİKLİ'ye,

Verdiği kıymetli bilgiler için Doç. Dr. Yıldıray YALMAN'a,

Cihaz destekleri ile gerçek platformda çalışmama olanak tanıyan, Microsoft Türkiye ve Sayın Mustafa KASAP'a,

Bu çalışma süresince desteklerini esirgemeyen ve sağladıkları huzurlu çalışma ortamından ötürü Kocaeli Üniversitesi Bilgisayar Mühendisliği Bölümündeki hocalarıma,

Yoğun çalışmalarım sırasında her zaman destek, anlayış ve fedakârlığını gördüğüm sevgili eşim Güler BAYRAKTAR UÇAR'a teşekkür ediyorum.

Temmuz – 2014 Ali UÇAR

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR i
İÇİNDEKİLER ii
ŞEKİLLER DİZİNİ iii
TABLOLAR DİZİNİ iv
SİMGELER DİZİNİ VE KISALTMALAR v
ÖZET vi
ABSTRACT vii
GİRİŞ 1
1. GÖRÜNTÜ DAMGALAMADA KULLANILAN YÖNTEMLER 4
1.1. Piksel Uzayında Görüntü Damgalama 4
1.2. Frekans Uzayında Görüntü Damgalama 7
1.2.1. AFD tabanlı görüntü damgalama 7
1.2.2. AKD tabanlı görüntü damgalama 8
1.2.3. ADD tabanlı görüntü damgalama 9
2. DONANIM VE YAZILIM 11
2.1. Donanım 11
2.1.1. Windows Phone 8 ile uygulama geliştirme ve görüntü İşleme 12
2.1.2. Windows Phone dosya sistemi (Windows Phone file system) 12
2.1.3. Tabii kod (Native code) 13
2.2. Yazılım 13
3. BULGULAR 20
4. SONUÇLAR VE ÖNERİLER 29
KAYNAKLAR 30
KİŞİSEL YAYIN VE ESERLER 33
ÖZGEÇMİŞ 34

ŞEKİLLER DİZİNİ

Şekil 1.1. Görüntü dosyalarının yapısı 5
Şekil 1.2. RGB Renk Modeli 5
Şekil 1.3. 8 Bitlik renk değeri 6
Şekil 1.4. LSB yönteminde 1 bitin yerleştirilmesi 6
Şekil 1.5. Değişiklik durumunda bit değerleri 7
Şekil 1.6. İki seviyeli ADD 9
Şekil 1.7. Dijital görüntü ve üç seviyeli ADD uygulanmış görüntü 9
Şekil 2.1. Nokia Lumia 820 akıllı telefonu 11
Şekil 2.2. Görüntülerin damgalanarak ekranda gösterimi 15
Şekil 2.3. Damgalama işlemi 15
Şekil 2.4. Dağılım 17
Şekil 2.5. Gömü verisi 18
Şekil 2.6. Damga tespiti 19
Şekil 3.1. Damga Çıkarım Başarım Grafiği 23
Şekil 3.2. Çekilen damgalı görüntüler 24
Şekil 3.3. Damga çıkarım doğruluk grafiği 28

TABLOLAR DİZİNİ

Tablo 2.1. Nokia Lumia 820'nin teknik özellikleri 11
Tablo 3.1. Dayanıklılık sabiti(α) değişimi sonucu PSNR değerleri..... 21
Tablo 3.2. Farklı α değerleri için damga çıkarımı..... 22
Tablo 3.3. Dayanıklılık sabiti değişimi ve damga çıkarım başarımı..... 23
Tablo 3.4. Damgalı görüntüye çeşitli saldırıların uygulanması sonrası görüntüler ... 25
Tablo 3.5. Damga doğrulukları 1 26
Tablo 3.6. Damga doğrulukları 2 27

SİMGELER DİZİNİ VE KISALTMALAR

α : Dayanıklılık sabiti değişkeni

Kısaltmalar

AC : Alternating Current (Alternatif Akım)
ADD : Ayrık Dalgacık Dönüşümü
AFD : Ayrık Fourier Dönüşümü
AKD : Ayrık Kosinüs Dönüşümü
BMP : Windows Bitmap
DC : Direct Current (Doğru Akım)
DCT : Discrete Cosine Transform (Ayrık Kosinüs Dönüşümü)
DFT : Discrete Fourier Transform (Ayrık Fourier Dönüşümü)
DICOM: Dijital Imaginig and Communications in Medicine (Tıbbi Dijital Görüntü ve İletişim)
DWT : Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)
EÖB : En Önemsiz Bit
FPGA : Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizileri)
FPS : Frame Per Second (Saniyedeki Frame Sayısı)
IGS : İnsan Görme Sistemi
IS : Isolated Storage (Ayrılmış Hafıza)
JPEG : Joint Photographic Experts Group (Birleşik Fotoğraf Uzmanları Grubu)
LSB : Least Significant Bit (En Önemsiz Bit)
MAX : Maximum (Maksimum)
MSE : Mean Square Error (Ortalama Karesel Hata)
PSNR : Peak Signal-to-Noise Rate (Tepe Sinyal Gürültü Oranı)
RGB : Red Green Blue (Kırmızı Yeşil Mavi)
SURF : Speeded Up Robust Feature (Hızlandırılmış Dayanıklılık Özelliği)
TAKD : Ters Ayrık Kosinüs Dönüşümü

GEZGİN ORTAMDA GÖRÜNTÜ DAMGALAMA UYGULAMASI

ÖZET

Günümüzde akıllı telefonlardan çekilen görüntülerin doğrudan cihaz üzerinden internette paylaşılması özellikle gençler arasında oldukça yaygın kullanılır hale gelmiştir. Fakat bu görüntüler internete yüklendikten sonra izinsiz olarak kopyalanabilmekte ve kullanılabilmektedir. Burada bir telif hakkı sorunu ortaya çıkmaktadır. İşte bu sorunun çözümü için, akıllı telefonda fotoğraf çekerken, fotoğrafın içerisine görünmez veri gizleyen (damgalayan) bir uygulama geliştirilmiştir. Damgalama işlemi için Ayrık Kosinüs Dönüşümü (AKD) esaslı, görünmez ve dayanıklı bir metot kullanılmıştır. Çeşitli saldırılara karşı içerdiği gizli bilgiyi muhafaza etme yeterliliği test edilmiş ve raporlanmıştır. Uygulama Windows Phone 8 işletim sistemine sahip, Nokia Lumia 820 üzerinde gerçekleştirilmiştir.

Anahtar Kelimeler: Akıllı Telefon, Ayrık Kosinüs Dönüşümü (AKD), Görüntü Damgalama, Steganografi, Windows Phone 8.

APPLICATION OF IMAGE WATERMARKING ON MOBILE PLATFORM

ABSTRACT

Nowadays, it has been very common among young people that photographs taken by smart phones are shared directly on the internet via the device. However, these images can be copied or misused without permission after being uploaded on the internet. At this point, the issue of copyright comes up. For the solution of this problem, an application that conceals invisible data into the image(watermarking) has been developed. A Discrete Cosine Transform based, invisible and robust method has been used. The application's capability to protect the concealed data from various attacks has been tested and briefly reported. The application has been operated on Nokia Lumia 820, which has Windows Phone 8 operating system.

Keywords: Smart Phone, Discrete Cosine Transform (DCT), Image Watermarking, Steganography, Windows Phone 8.

GİRİŞ

Hayatımızın her alanında yer alan, pazara sunulduğu andan itibaren hızla yaygınlaşan kişisel bilgisayarlar ile vazgeçilmez iletişim aracı olan cep telefonlarının, yüksek çözünürlüklü kameraların bir araya gelmesi ile ortaya çıkan akıllı telefonlar son yılların en önemli teknolojik gelişmelerinden biridir. Akıllı telefonlar, boyutlarının küçülmesi, performansının artması, erişilebilir fiyatlara ulaşması ile birlikte hızla yaygınlaşmıştır. Bu durumu internet ve kablosuz haberleşme alanındaki gelişmelerin takip etmiştir. Bu sayede sayısı hızla artan akıllı telefon kullanıcıları birçok dijital platforma erişim imkânı buluyor.

Günümüzde akıllı telefonlardan en yaygın kullanılan dijital platformlardan biri hiç şüphesiz sosyal medya platformlarıdır. İnsanlar günlük hayatlarında akıllı telefonlarından fotoğrafladıkları görüntüleri ve pek çok bilgiyi sosyal medya üzerinden paylaşmaktadırlar. Fakat bu resimler internete yüklendikten sonra izinsiz olarak kopyalanabilmekte, kullanılabilmektedir. Bazen de sahibi tarafından internete yüklenmeyen resimler casus yazılımlar ve kötü niyetli kişiler tarafından kopyalanabilmekte ve istenildiği gibi kullanılabilmektedir. Burada bir telif hakkı sorunu ortaya çıkmaktadır. İşte bu sorunun çözümü için, akıllı telefonda fotoğraf çekerken görüntü içerisine görünmez veri gizleyen bir uygulama geliştirilmiştir.

Görüntü içerisine veri gizleme çalışmaları stenografi ve görüntü damgalama olarak adlandırılır. Görüntü stenografi çalışmalarında amaç gizli bir bilginin taşıyıcı bir resim üzerine gömülerek taşınmasıdır. Taşıyıcı resim üzerindeki bilginin fark edilmemesi ve görüntü üzerine maksimum veri gömme stenografi çalışmalarında büyük öneme sahiptir. Görüntü damgalama çalışmalarında ise amaç resim üzerine gömülen verinin yani damganın çeşitli saldırılara karşı dayanıklı olmasıdır. Fark edilmezlik ve taşıyıcı görüntü üzerine eklenebilecek maksimum veri, görüntü damgalama çalışmaları için önemli olsa da ikinci planda yer alır. Asıl önemli olan damganın dayanıklılığıdır. Stenografi, damgalama ve bu alanda yapılan çalışmalar

ileriki bölümlerde detaylı olarak anlatılacaktır. Bu tezde dayanıklı görüntü damgalama metodu kullanılmıştır.

Akıllı telefon teknolojisi henüz yeni sayılabilecek bir gelişme olmasına rağmen araştırmacıların dikkatini çekmiş, sınırlı sayıda da olsa görüntü damgalama uygulamaları akıllı telefonlar üzerinde gerçeklenmiştir. Gerçekleştirilen uygulamalar kırılgan damgalama ve dayanıklı damgalama olarak iki bölüme ayrılabilir. Damgalanmış görüntü dışarıdan gelebilecek jpeg sıkıştırma, gürültü ekleme, yeniden boyutlandırma vb. saldırılara karşı belli oranlarda dayanıklı ise damgalama dayanıklı bir metot ile, dayanıklı değiş ise kırılgan bir metot ile yapılmış demektir. Android sistem üzerinde kaos teorisi tabanlı kırılgan bir görüntü damgalama uygulamasında, 512x512 boyutlarında bir görüntünün damgalanmasında geçen süre yaklaşık 3 sn civarında ölçülmüştür (Palma Hernandez ve Torres-Huitzil, 2011). Yine bu çalışmada, frekans uzayındaki dayanıklı görüntü damgalama uygulamalarının çok devasa işlemler gerektirdiğinden, mobil cihazların sınırlı kaynakları olduğu için bu tür uygulamalarda çok daha az hesaplama gerektiren bu metodun tüm cihazlar üzerinde daha uygulanabilir olduğu vurgulanmıştır. Başka bir çalışmada, İnternete görüntü upload edilmeden önce otomatik olarak damgalayan, Red,Green,Blue (RGB) renk değerlerini ayrıştırıp, G(yeşil) kanalında kaotik seri tabanlı veri gizleme işlemi yapan ve G kanalındaki değerlere göre damgayı messy algoritması ile oluşturan yine piksel uzayında görüntü damgalayan bir adroid uygulanması gerçeklenmiştir (Patil ve diğ., 2014).

Adroid sistem üzerinde, frekans dönüşümü (DCT) tabanlı hymnMark olarak adlandırılan görüntü damgalama uygulamasında Koch ve Zhao(1995)'nun algoritması üzerinde bazı değişiklikler ve optimizasyonlar yapılarak kullanılmış, ayrıca güç tüketim değerleri ve görüntü üzerine bazı saldırılar yapılarak dayanıklılık değerleri raporlanmıştır (Miao ve diğ., 2012). Cep telefonu kameralarından yakalanan görüntüler için önerilen bir çalışmada ise görüntüyü hem frekans dönüşümü tabanlı (DWT) ardından da piksel uzayında kırılgan damgalayan bir yöntem önerilmiş, ataklara karşı dayanıklılık sonuçları raporlanmış ama herhangi bir mobil platformda uygulama sonuçlarına değinilmemiştir (Jassim ve diğ., 2013). İnternete görüntü yükleme sırasında otomatik olarak görüntüyü yeniden

boyutlandıran ve dayanıklı damgalayan farklı bir çalışmada ise, Haar dalgacık dönüşümü temelli yaklaşım ile görüntüye veri gizlenmiştir (Chen ve Huang, 2011).

Bu tez çalışmasında Windows Phone işletim sistemine sahip, Nokia Lumia 820 akıllı telefonu üzerinde, fotoğraf çekerken, dayanıklı bir metotla görüntü damgalayan mobil bir uygulama geliştirilmiştir. Damgalanan görüntülerin dayanıklılığı, uygulama performansı test edilmiş açıkça sunulmuştur. Bu çalışma Windows Phone ile yapılan ilk dayanıklı görüntü damgalama uygulaması olmasının yanı sıra, fotoğraf çekerken ekranda görünen her bir görüntü karesi dâhil, damgalanmamış görüntünün hiçbir zaman var olmaması ve işlem performansı en önemli farklılığıdır.

Tezin birinci bölümünde görünmez görüntü damgalamada kullanılan yöntemler, piksel uzayında ve frekans uzayında kullanılan yöntemler olarak iki bölümde sunulmuş ve karşılaştırılmıştır. İkinci bölümde donanım ve yazılım olarak iki kısımda ele alınmış, donanım kısmında Windows Phone 8'den detaylı olarak bahsedilmiş, bu donanım ile görüntü işleme için gerekli bilgiler sunulmuştur. Yazılım kısmında ise damgalama metodu ve geliştirilen yazılım ayrıntılı bir şekilde anlatılmıştır. Üçüncü bölümde PSNR, damga çıkarım doğrulukları, damgalanmış görüntüler ve onlara yapılan saldırılar sonrası elde edilen bulgular açıkça verilmiştir. Dördüncü sonuç bölümünde ise çalışma değerlendirilmiş ve ileriki zamanlarda yapılabilecek çalışmalara değinilmiştir.

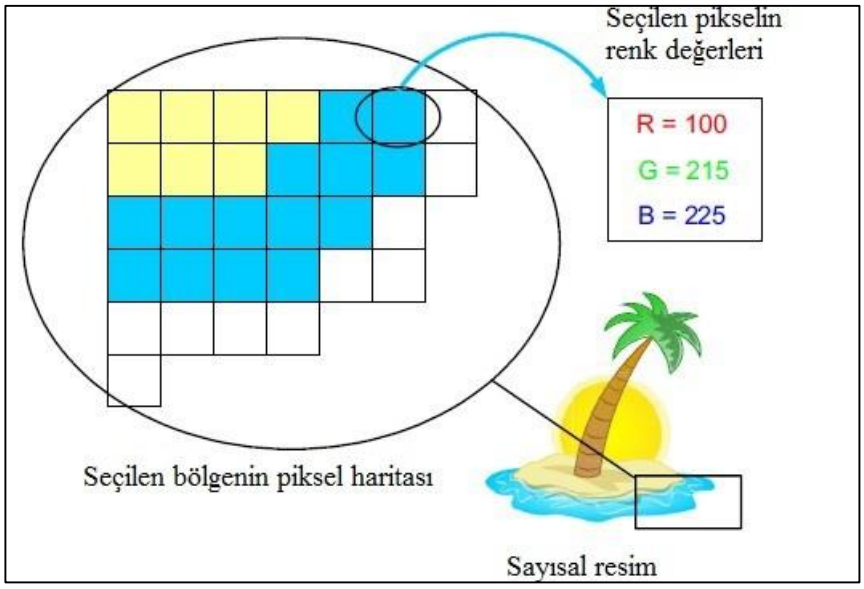
1. GÖRÜNTÜ DAMGALAMADA KULLANILAN YÖNTEMLER

Görünmez görüntü damgalama uygulamalarında kullanılan yöntemler piksel uzayında görüntü damgalama ve frekans uzayında görüntü damgalama olmak üzere iki bölümde incelenebilir.

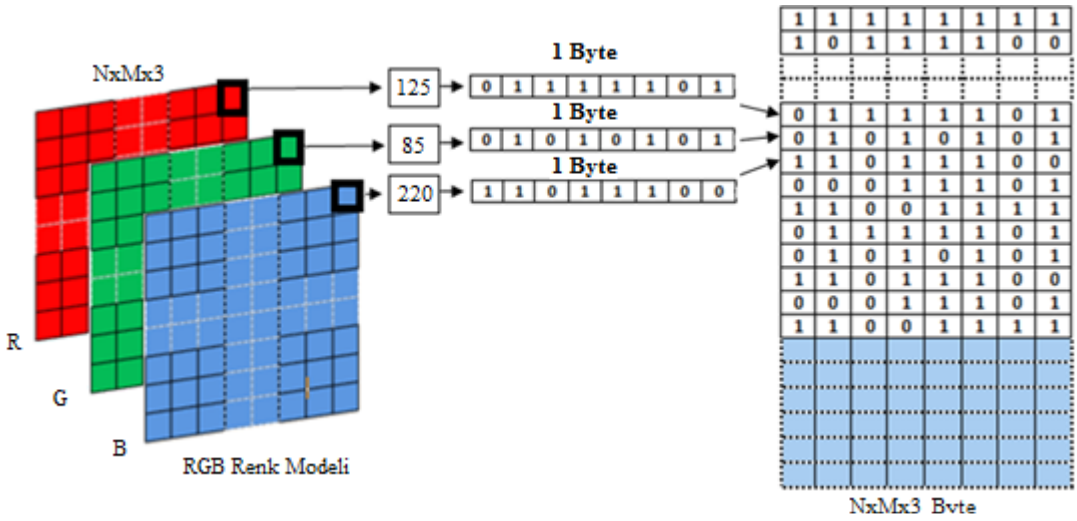
1.1. Piksel Uzayında Görüntü Damgalama

Piksel uzayında görüntü damgalama konusu literatürde hem watermarking (damgalama) hem de steganografi (veri gizleme veya sırrını örtme) uygulamalarına konu olmuştur. Bazı araştırmacılar bu yöntemi damgalama başlığı altında ele alırken, bazıları da stenografi başlığı altında ele almaktadır. Steganografi kelimesi örtülü yazı anlamına gelen, yunan alfabesinden türemiş bir kelimedir (Cummins ve diğ., 2004). Stenografi de amaç fark edilemezlik ve fazla verinin taşıcıya yüklenmesidir. Damgalamada ise amaç taşıyıcı üzerindeki damganın dayanıklılığı ve orijinal görüntü ile kıyaslandığında damgalanmış görüntüdeki bozulmanın kabul edilebilir düzeyde olmasıdır.

Akıllı telefonlar günümüzde yüksek performans ve hızlara ulaşmıştır. Ancak işlemci hızı ve bellek açısından bilgisayarlarla kıyaslandığında yine de sınırlı kapasiteye sahip oldukları söylenebilir. Bu nedenle yapılacak uygulamaların tasarruflu olması gerekir. Ancak görüntü işleme işlemleri için tasarruftan bahsetmek pek mümkün değildir. RGB renk modelinde bir piksel kırmızı,yeşil ve mavi olmak üzere 3 tane 8 bitlik sayısal değer ile ifade edilir (Şekil 1.1). NxM boyutlarındaki bir görüntünün tümü üzerinde yapılacak küçük bir değişiklik için bile, yapılacak işlem sayısı NxM adet kullanılacak bellek ise yaklaşık NxMx3 byte kadardır (Şekil 1.2).

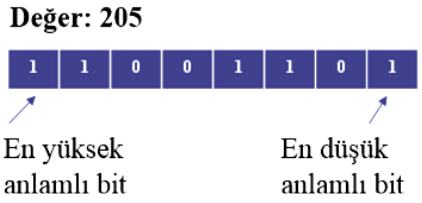


Şekil 1.1. Görüntü dosyalarının yapısı (Çetin ve Özcerit, 2009)



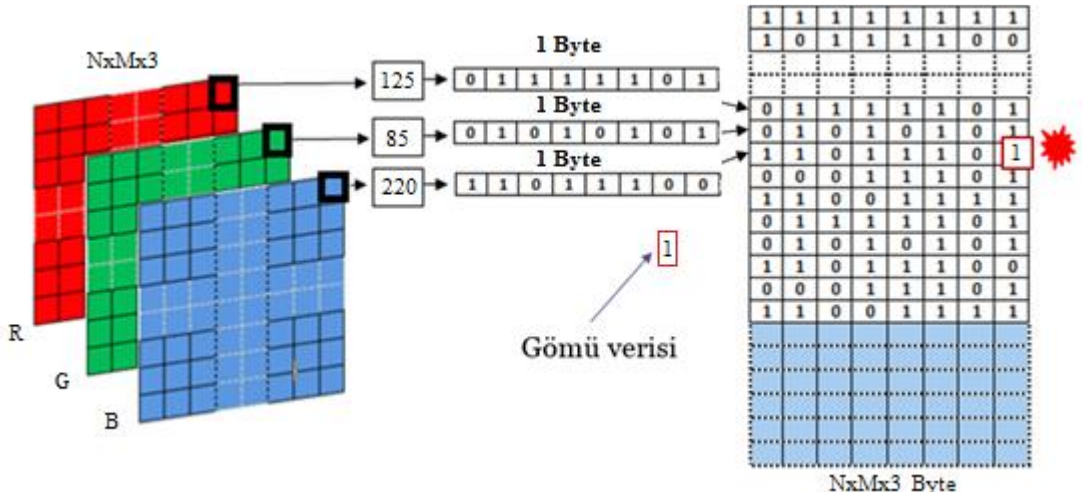
Şekil 1.2. RGB Renk Modeli

Piksel uzayında görüntü damgalama yöntemlerinden, en bilineni Least Significant Bit (En Önemsiz Bit) görüntü damgalamadır. Bu yöntem pikselin en düşük anlamlı bit değerinin (Şekil 1.3), yerleştirilecek veri ile yer değiştirmesi esasına dayanır (Şekil 1.4). LSB yönteminin uygulanması oldukça kolay ve orijinal görüntüdeki bozulma çıplak gözle fark edilemeyecek kadar az olduğu için sıklıkla tercih edilmiştir (Simmons, 1984).



Şekil 1.3. 8 Bitlik renk değeri

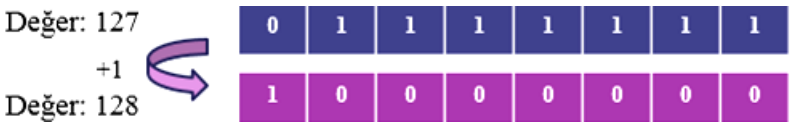
Akleylek ve Nuriyev (2005), geliştirdikleri açık anahtar ve gizli anahtar çiftiyle çalışan, AS knapsack ismini verdikleri şifreleme sistemi ile resim dosyalarının en az önemli bitlerini değiştirme yöntemi sayesinde veri saklayan bir yöntem önererek, alıcıya yollanan verinin inkar edilmemesi ve alıcının göndericinin ilettiği gerçek veriyi görebilmesi sağlanmıştır. Lee ve Chen (2000), gri resimlerde LSB yöntemiyle piksel değerini oluşturan bitlerin ilk dördünün değiştirilmesi ile, %50 ye yaklaşan kapasiteyle veri saklamışlardır. Shahreza (2006), mobil telefonlarda SMS (Short Message Service) yoluyla ve siyah beyaz görüntüler içerisine bilgi saklayarak gizli mesajlaşmayı sağlanmayan bir öneri sunmuştur.



Şekil 1.4. LSB yönteminde 1 bitin yerleştirilmesi

En düşük anlamlı bir yerine daha yüksek anlamlı bir bit değeri gömü verisini yerleştirmek için seçilirse, damga dayanıklılığı açısından daha dayanıklı bir sonuç elde edilip edilmeyeceği ise şu şekilde değerlendirilebilir: RGB renk modelinde R(Kırmızı) değerine gömü verisi yerleştirilmiş bir görüntünün, gömü verisi yerleştirilmiş bir pikselin kırmızı renk değeri R=127 olsun. İnsan gözü tarafından fark edilmeyecek, +1 değerlik bir bozulma uygulanırsa renk değeri R=128 olur. Şekil 1.5 'te 127 ve 128 değerlerinin 8 bitlik ifadeleri verilmiştir. Görüldüğü gibi çok

küçük bir değişiklikte 7/8 piksel değeri değişmiştir. Bu nedenle piksel uzayında yapılan damgalamalar çoğu zaman dayanıksızdır.



Şekil 1.5. Değişiklik durumunda bit değerleri

RGB, YUV, CMYK gibi renk modellerindeki renk katmanlarında bit düzeyinde işlem yaparak gömü verisinin belirlenmiş bazı bitlere belli sıralarla yerleştirilmesi yöntemi çok sayıda stenografi çalışmasına konu olmuştur.

1.2. Frekans Uzayında Görüntü Damgalama

Frekans uzayında görüntü işlemenin esası matematiksel bazı dönüşümler aracılığı ile gömü verisinin frekans uzayındaki uygun bileşenlere eklenmesi esasına dayanır. Bu amaçla AKD(Ayrık kosinüs dönüşümü), AFD(Ayrık Fourier Dönüşümü), ADD(Ayrık Dalgacık Dönüşümü) tabanlı yöntemler kullanılmaktadır.

1.2.1. AFD tabanlı görüntü damgalama

AFD tabanlı görüntü damgalamada görüntüye 2 boyutlu AFD uygulanarak AFD katsayıları elde edilir. Katsayılara gömü verisi yerleştirildikten sonra ters dönüşüm yapılarak damgalanmış görüntü elde edilir. Li ve arkadaşları(2012) bu yöntemi tıbbı görüntüler üzerinde uygulamış ve düşük frekans katsayılarına logistic map yöntemiyle şifrelenen gömü verisini yerleştirmiş, pek çok saldırıya dayanıklı olduğunu belirtmişlerdir. Yine tıbbi görüntüler üzerinde yapılan başka bir çalışamada,standart DICOM(Digital Imaging and Communications in Medicine) formatındaki tıbbi görüntülerin damgalanmasında AFD tercih edilmiş, gömü verisi orta frekans katsayılarına yerleştirilmiş , farkedilmezlik ve dayanıklılık açısından iyi sonuçlar elde edilmiştir (Hernandez ve diğ., 2012a). Speeded Up Robust Feature(SURF) algoritması ve AFD dönüşümünün birlikte kullanıldığı başka bir çalışmada ise yine gömü verisi için orta frekans katsayılar tercih edilmiş, geometrik saldırılardan etkilenmeyen ve pek çok saldırıya karşı dayanıklı SURF-AFD tabanlı bir yöntem geliştirmişlerdir (Hernandez ve diğ., 2012b).

1.2.2. AKD tabanlı görüntü damgalama

Ayrık kosinüs dönüşümü hesaplama ve sıkıştırma verimliliği, frekans uzayındaki uygun enerji dağılımından dolayı pek çok görüntü ve video kodlama sisteminde tercih edilmiştir ki buna JPEG, MPEG, H.26x örnek verilebilir (Lu ve diğ., 2010). Öyle ki bu kodlama standartlarıyla bağdaşmıştır (Chen, 2006). Bu nedenle dayanıklı görüntü damgalama çalışmalarında da yaygın olarak tercih edilir. Ayrık kosinüs dönüşümü tabanlı görüntü damgalama uygulamasının temeli aşağıda verilmiştir:

1. Görüntü 8x8 bloklara ayrılarak her bir bloğa AKD uygulanır.
2. Her bir bloktan elde edilen 8x8 AKD katsayılarının bazılarında değişiklik yapılır.
3. Ters AKD uygulanarak damgalanmış görüntü elde edilir.

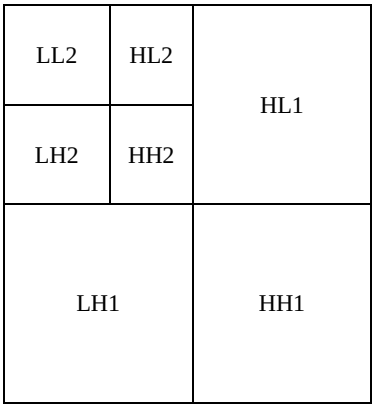
Ayrık kosinüs dönüşümü tabanlı görüntü damgalama yöntemi ileriki bölümlerde daha detaylı anlatılacaktır.

Damgalanan bir görüntünün kasıtlı veya kasıtsız uğrayabileceği saldırılardan biri JPEG sıkıştırma saldırısıdır. Herhangi bir damgalama yönteminin dayanıklılığını test etmek için, median filter, salt and paper, gaussian filter gibi saldırılar da uygulanır ama bu saldırılar sonrası görüntüde gözle görülür bozulmalar olur ve orijinal kaliteden oldukça uzaktır. Ama JPEG sıkıştırma sonrası gözle görünür ciddi bir bozulma olmaz ve her ne kadar kayıplar olsa da, insan görme sistemi tarafından fark edilemez ve orijinal kaliteye çok yakın görüntüler elde edilir. Bu nedenle damgalanan görüntünün öncelikle JPEG sıkıştırmaya karşı dayanıklı olması beklenir. Günümüzde mobil cihazlar, internet ve kablosuz iletişim araçları oldukça gelişmiş olmasına rağmen, hala sınırlı miktarda kaynağa sahip olduğu söylenebilir.

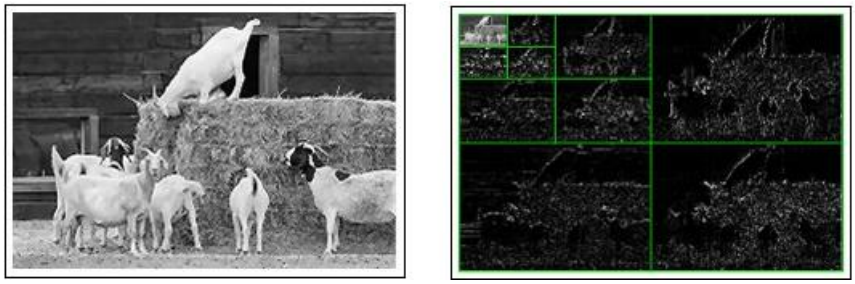
Bitmap görüntüler JPEG görüntülere göre –sıkıştırma oranına bağlı olarak- daha büyük boyutlara sahiptir. Çoğu akıllı telefonda standart kameradan resmi Bitmap olarak kaydetmek yer almamaktadır. Akıllı telefon kameraları gün geçtikçe gelişmekte, daha yüksek kalitede fotoğraf çekebilir hale gelmektedir. Kameranın yakaladığı yüksek çözünürlüklü görüntünün boyutu, mobil platformun kullanım amacı düşünüldüğünde Jpeg kodlamanın neden yaygın olduğu ve Bitmap kodlamanın neden yer almadığı daha rahat anlaşılır.

1.2.3. ADD tabanlı görüntü damgalama

Ayrık dalgacık dönüşümdeki temel amaç işaretin içerisindeki yüksek ve düşük frekans alanlarının ayrıştırılmasıdır. ADD işareti dört ayrı frekans alt bandına (Low-low, low-high, high low ve high-high) ayrıştırır ve bu bir seviyeli ADD'dir (Mehta ve diğ., 2013). Alt bantlara gerekli görülürse tekrar ADD uygulanarak kendisini temsil eden dört ayrı alt banda ayrıştırılabilir ve iki seviyeli ADD uygulanmış olur, alt bantlara ayırma işlemi gerekli görüldüğü kadar devam ettirilebilir (Şekil 1.6 ve Şekil 1.7). Yapılan çalışmalarda alçak frekans bantları, orta ve yüksek frekans bantlarına göre daha yaygın kullanıldığı görülse de, farklı ADD frekans bantlarının seçildiği çalışmalar da mevcuttur (Kundur ve Hatzinakos, 1997) (Barni ve diğ., 1999). Her görüntü verisi birbirinden farklı olduğu için, aynı frekans alt bandı seçimi, aynı damga veya gömü verisi ekleme ve çıkarma parametreleri her görüntü için benzer dayanıklılıkta ve benzer PSNR değerinde optimum sonuçlar vermeyebilir. Bu nedenle uygun frekans bantlarının ayrıştırılmasında, damganın çıkarılmasında gerekli olan eşik değerinin hesaplanmasında, uygun bloklarının seçilmesinde pek çok yöntem ADD ile birlikte kullanılmıştır.



Şekil 1.6. İki seviyeli ADD



Şekil 1.7. Dijital görüntü ve üç seviyeli ADD uygulanmış görüntü (URL-1)

Piksel uzayında yapılan görüntü damgalama uygulamaları kırılgan bir yapıya sahip olmasına rağmen, frekans uzayında yapılan görüntü damgalama uygulamalarına göre daha fazla veri taşıyıcı görüntü üzerine gömülebilir. Ayrıca uygulaması daha kolaydır. Daha az hesap gerektirdiği için de daha performanslı bir şekilde uygulanabilir. Frekans uzayında yapılan görüntü damgalama uygulamalarında ise, frekans dönüşümü belirli bir bloğa uygulandığı için ve her bir frekans bileşenine gömü verisi yerleştirilemediği için taşıyıcı görüntü üzerine, piksel uzayında yapılan görüntü damgalamaya göre daha az veri gömülebilir. Piksel uzayında yapılan görüntü damgalama uygulamalarına göre çok daha fazla hesap gerektirdiği için daha yavaştır. Ancak frekans dönüşüm temelli metotlar piksel uzayındaki yöntemlere göre daha dayanıklıdır.

2. DONANIM VE YAZILIM

2.1. Donanım

Bu tez çalışmasında görüntü damgalama uygulamasının gerçekleneceği platform olarak, Windows Phone 8, cihaz olarak ise Nokia Lumia 820(Şekil 2.1) seçilmiştir. Nokia Lumia 820'nin teknik özellikleri Tablo 2.1'de yer almaktadır.



Şekil 2.1. Nokia Lumia 820 akıllı telefonu

Tablo 2.1. Nokia Lumia 820'nin teknik özellikleri

Kamera	Carl Zeiss lens, Ana kamera sensör boyutu: 8,7 MP 16:9 sensör
İşlemci	Qualcomm Snapdragon™ S4 , Çift çekirdekli 1,5 Ghz
Boyutlar	Yükseklik: 123,8 mm Genişlik: 68,5 mm Kalınlık: 9,9 mm Ağırlık: 160 g
Sensörler	Ortam ışığı sensörü, İvmeölçer, Yakınlık sensörü, Cayroskop, Manyetometre
Hafıza	Kullanıcı verileri depolama: Cihaz İçi, Hafıza kartı RAM: 1 GB Yığın hafıza: 8 GB Genişletilebilir hafıza kartı türü: MicroSD En fazla hafıza kartı boyutu: 64 GB Ücretsiz bulut depolaması: 7 GB
İşletim Sistemi	Windows Phone 8

2.1.1. Windows Phone 8 ile uygulama geliştirme ve görüntü İşleme

Windows Phone 8 ile uygulama geliştirmek için Visual Studio 2012 veya Visual Studio 2013 geliştirme ailesinden birine sahip olmak gerekir. Eğer 2012 versiyonu ile geliştirme yapılacak ise, Windows Phone SDK 8.0 ın yüklenmesi gerekir. 2013 versiyonunda ise bu hazır gelmektedir.

Windows Phone 8 de, kamera ve dosya sınıfları kameradan alınan görüntüyü sadece Jpeg olarak telefonun Medya kütüphanesine veya Isolated Storage a kaydedilebilecek şekilde tasarlanmıştır. Görüntü işleme uygulamalarında ise görüntü üzerinde yapılan değişikliklerin ve kodun %100 doğru olduğundan emin olmak için Bitmap formatında kaydetmeye ihtiyaç duyulur.

Eğer piksel uzayında Windows Phone 8 platformunda kırılgan bir metot ile görüntü damgalama uygulaması yapılacaksa, bitmap dosya formatında kaydetme işleminin de kodlanması gerekir. Daha detaylandırılacak olursa, bitmap header file, verilerin dosyaya belirli bir düzen içerisinde yazılması işlemi yazılımcı tarafından kodlanmalıdır.

WP8 aynı zamanda bitmap formatındaki resmi Medya galerisine (MediaLibrary) bitmap formatında kaydetmeye izin vermez, sadece Jpeg formatında resim medya galerisine kaydedilebilir. Bu nedenle IS'e kaydedilen resimler, telefonun fotoğraf görüntüleyicisi tarafından görüntülenemez.

2.1.2. Windows Phone dosya sistemi (Windows Phone file system)

Windows Phone da masaüstü windows işletim sistemlerinden farklı olarak kullanıcıların bütün dosya sistemine girmelerine izin verilmez. Burada kullanıcıları kötü niyetli uygulamalardan korumak amaçlanmıştır. Windows Phone üzerine yüklenen her uygulama yalnızca kendisine ayrılan ve isolated storage olarak adlandırılan hafızaya System.IO.IsolatedStorage namespace altındaki class ları kullanarak erişebilir. Örneğin, "Images/Picture/NewImage.jpeg" şeklinde dosya yolu belirtilerek kayıt yapmak, uygulama için ayrılan isolated storage için mümkündür.

Ayrıca IS'e kaydedilmiş resimler veya dosyalar telefonun görüntüleyicileri tarafından görüntülenemez. Örneğin, IS'e "deneme.bmp" bir resim kaydedildiğini

varsayalım. Telefondan medya galerisine girildiğinde telefon kamerası tarafından daha önce çekilen tüm resimler görüntülenebilirken, "deneme.bmp" adlı resim bulunamaz. Isolated Storage daki dosyalara erişmenin iki yolu vardır: Birincisi, yazılımcı kod tarafından erişilebilir ve is üzerindeki dosyalar görüntülenebilir. İkincisi ise IS deki dosyaları görüntülemek amacıyla hazırlanmış, özel uygulamalar kullanılabilir. Bu uygulamalardan birisine "Windows Phone Power Tools" örnek verilebilir.

2.1.3. Tabii kod (Native code)

Windows Phone 8, Visual C++ 2012 destekler ve bu sayede C/C++ dilinde yerel kod(native code) ile uygulama geliştirmeye imkân sağlar. Tüm Windows Phone 8 uygulamaları (hepsinin native yani yerel koda ihtiyacı olamasa da) yerel kod kullanabilecek şekilde dizayn edilmiştir (URL-2). Native code kullanım sebepleri şu şekilde sıralanır:

- ✓ Çoklu platform kullanımı hedefleniyorsa, uygulamanın çekirdeği native code kütüphanesi olarak yazılarak, tüm platformlarda kullanılabilir.
- ✓ Eğer uygulama ekranda grafik işlemleri için Direct3D'ye ihtiyaç duyuyorsa, Native code kullanımı zorunludur.
- ✓ Native kullanılarak büyük hesaplama gerektiren işlemlerde performans kazanımı sağlanabilir. Native code ile bir üst katmanda yer aldığı söylenebilecek maneged kod arasındaki genel ilişkiden ötürü, büyük hesaplama gerektiren işlemlerde managed code kullanımı yerine native code kullanmanın peşinen daha performanslı olduğu düşünülmemelidir.

Windows Phone 8 uygulamaları aynı zamanda, mobil masaüstü uygulamaları için Windows Runtime kullanımı da mümkündür. Projenin yanında ayrıca eklenecek bir Windows Runtime Compenent projesi ile gerçek zamanda (Run Time) kamera, ses-konuşma, network, konum vb gibi özelliklere erişilebilir (URL-2).

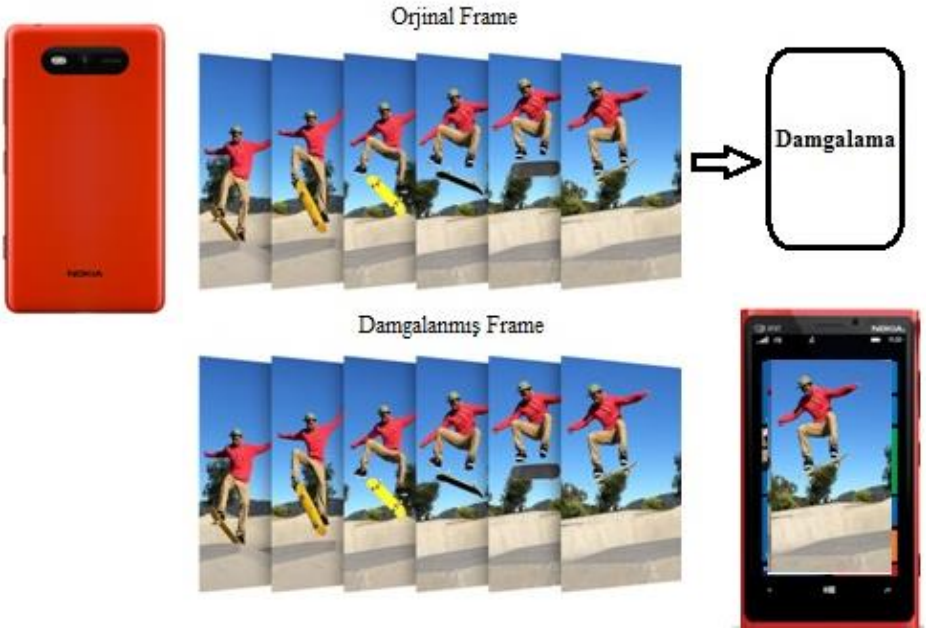
2.2. Yazılım

Dijital dünyanın kullanıcıları her zaman pek çok güvenlik problemi ile karşı karşıyadır. İnsanlar internet üzerindeki hesaplarına, bilgisayarlarına güçlü şifreler kurarak, pahalı yazılımlar kullanarak her türlü kişisel alanlarına yetkisiz erişimi

engelleme çalışırlar. Alınan her türlü önleme rağmen, %100 güvenlikten bahsetmek oldukça zordur. Doğası gereği kolayca kopyalanabilen dijital veriler, internet üzerinden paylaşılmasa bile, cihazlardan doğrudan kopyalanmak suretiyle başkalarının eline geçebilir. İşte bu noktadan hareketle görüntüler için orijinal görüntünün, yani damgasız görüntünün hiçbir zaman var olmaması prensibine dayalı, mobil platformda dayanıklı yöntemlerle damgalayarak fotoğraf çeken bir uygulama geliştirilmiştir. Her ne kadar bellekte damgalanmamış görüntü kareleri ham halde kısa süre yer alsalar da, ekranda damgalanmış halleri gösterilir ve telif hakkı yerleştirilmemiş yani damgalanmamış görüntünün hiçbir zaman var olmamasından kasıt budur. Damgalanmış görüntüden damga orijinal görüntüye ihtiyaç duyulmaksızın çıkarılabilir.

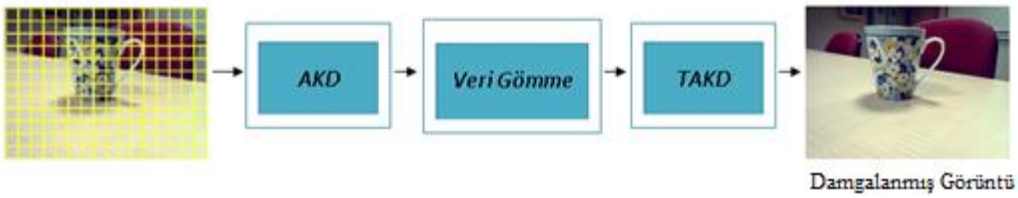
Kamera uygulamaları çalıştırıldığında, belirli aralıklarla kameradan görüntü alınarak ekranda video olarak gösterilir. Kullanıcı fotoğraf çekme butonuna bastığında ise o anki video karesi resim olarak kaydedilir. Bu tez çalışmasında gerçekleştirilen uygulamada, fotoğraf çekme sırasında ekranda görüntülenen her bir video karesi damgalanarak görüntülenir (Şekil 2.2) ve fotoğraf çekme butonuna basılarak yakalanan görüntü de bu video karelerinden biridir.

Damgalama renkli görüntünün, RGB renk modelinde Blue (mavi) katmanı üzerinden gerçekleştirilir. Green (yeşil) ve Red (Kırmızı) katmanları üzerinde herhangi bir işlem yapılmaz. Damgalama işlemi için AKD tabanlı dayanıklı bir metot kullanılmıştır.



Şekil 2.2. Görüntülerin damgalanarak ekranda gösterimi

Gömü verisi yerleştirme veya damgalama işlemi için 640x480 boyutlarında her bir video karesinin mavi katmanı 8x8 lik bloklara ayrılır. Her bir bloğa AKD uygulanır. Sonuçta 8x8 adet AKD frekans bileşeni elde edilir. Her 8x8 lik bloğa 1 bit veri yerleştirilir. Bu nedenle 640x480 boyutlarındaki görüntü için yerleştirilebilecek maksimum gömü verisi, 4800 bittir. Gömü verisi yerleştirme işlemi yapıldıktan sonra nicemleme matrisi ile nicemlenir ve TAKD uygulanarak damgalanmış görüntü elde edilmiş olur.



Şekil 2.3. Damgalama işlemi

2 Boyutlu AKD formülü Denklem (2.1) de, TAKD formülü Denklem (2.2) de verilen (Jain, 1989);

$$F(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{2.1}$$

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)F(u,v)\cos\left(\frac{(2x+1)u\pi}{2N}\right)\cos\left(\frac{(2y+1)v\pi}{2N}\right) \tag{2.2}$$

şekilde hesaplanır. x,y,u,v=0,1,2,…,7; f(x,y) kaynak görüntünün piksel değerleri, F(u,v) AKD katsayılarıdır.

Denklem (2.1)'in matris formatı Denklem (2.4) deki gibi, Denklem (2.2)'nin matris formatı Denklem (2.5) deki gibi ifade edilebilir. F matrisinin gösterimi Denklem (2.6), f matrisinin gösterimi Denklem (2.7), G matrisinin gösterimi ise Denklem (2.8) deki gibi;

$$F = G.f.G^T \tag{2.4}$$

$$f = G^T.F.G \tag{2.5}$$

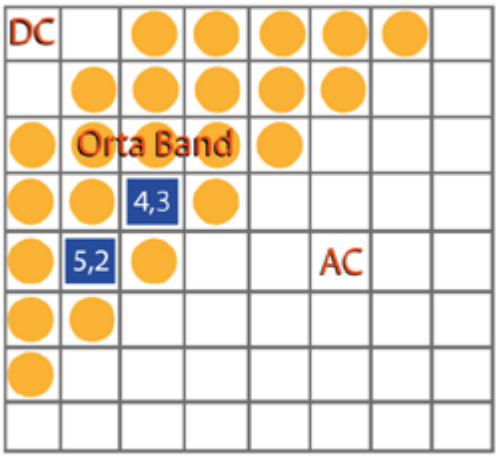
$$F = \begin{bmatrix} F(0,0) & F(0,1) & \cdots & F(0,7) \\ F(1,0) & F(1,1) & \cdots & F(1,7) \\ \vdots & \vdots & \cdots & \vdots \\ F(7,0) & F(7,1) & \cdots & F(7,7) \end{bmatrix} \tag{2.6}$$

$$f = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,7) \\ f(1,0) & f(1,1) & \cdots & f(1,7) \\ \vdots & \vdots & \cdots & \vdots \\ f(7,0) & f(7,1) & \cdots & f(7,7) \end{bmatrix} \tag{2.7}$$

$$G = \frac{1}{2}\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}} \\ \cos\frac{\pi}{16} & \cos\frac{3\pi}{16} & \cdots & \cos\frac{15\pi}{16} \\ \vdots & \vdots & \cdots & \vdots \\ \cos\frac{7\pi}{16} & \cos\frac{21\pi}{16} & \cdots & \cos\frac{105\pi}{16} \end{bmatrix} \tag{2.8}$$

ifade edilebilir.

JPEG gibi görüntü sıkıştırma işlemleri yüksek frekans katsayılarına etki eder. Bu nedenle düşük frekans katsayılarına yerleştirilecek veri, diğer bileşenlere göre sıkıştırma işlemlerine maruz kaldığında daha kolay sinebilir. Alçak frekans bileşenlerinde yapılacak değişikliklere ise IGS oldukça duyarlıdır ve değişikliğin fark edilme ihtimali daha fazladır (Huang ve diğ., 2000). Şekil 2.4'te 8x8 boyutlarında matrisin DCT katsayıları ve seçilen katsayılar yer almaktadır.



Şekil 2.4. Dağılım (Yeşilyurt ve diğ., 2012)

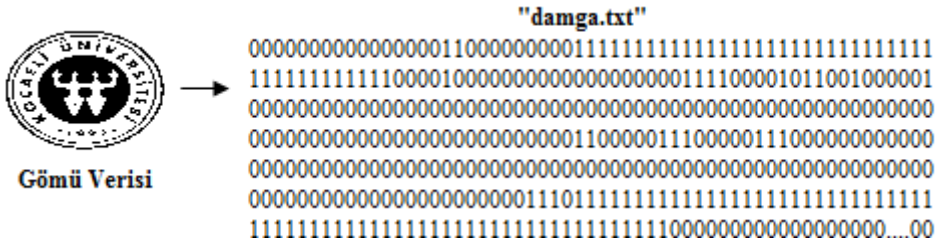
Bu çalışmada gömü verisi yerleştirmek için orta frekans bandındaki (4,2) (5,3) nolu AKD katsayısı seçilmiştir. Gömü verisi 1 ise (5,2) de bulunan AKD frekans katsayısı, (4,3) de bulunan AKD katsayısı bileşeninden büyük olması gerekir. Değilse yer değiştirilir. Eğer gömü verisi 0 ise (4,3) de bulunan AKD frekans katsayısı, (5,2) de bulunan AKD frekans katsayısından büyük olması istenir, değilse yer değiştirilir.

Seçilen frekans bileşenleri arasındaki fark az olduğundan bir dayanıklılık sabiti (α) tanımlanmıştır ve büyük olacak olan değere eklenir. Dayanıklılık sabiti değeri için farklı değerler ile yapılan denemeler ve sonuçları bulgular bölümünde sunulmuştur.

Gömü verisi yerleştirme sonrasında nicemleme (quantization) işlemi yapılır. Gömü verisi yerleştirilmiş 8x8 lik AKD bloğu, Eşitlik (2.9) da verilen nicemleme matrisi ile nicemlenir;

$$Q = \begin{bmatrix} 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 \\ 5 & 7 & 9 & 11 & 13 & 15 & 17 & 19 \\ 7 & 9 & 11 & 13 & 15 & 17 & 19 & 21 \\ 9 & 11 & 13 & 15 & 17 & 19 & 21 & 23 \\ 11 & 13 & 15 & 17 & 19 & 21 & 23 & 25 \\ 13 & 15 & 17 & 19 & 21 & 23 & 25 & 27 \\ 15 & 17 & 19 & 21 & 23 & 25 & 27 & 29 \\ 17 & 19 & 21 & 23 & 25 & 27 & 29 & 31 \end{bmatrix} \tag{2.9}$$

Damga verisi olarak, siyah-beyaz ikili renkte 80x60 boyutlarındaki görüntü seçilmiştir (Şekil 2.5). Siyah olan pikseller "1" beyaz olanlar ise "0" olacak şekilde damga verisi bir text dosyasına yazılarak uygulamanın gerçeklendiği Nokia Lumia 820'ye yüklenmiştir. Damgalayarak fotoğraf çekme uygulaması çalıştırıldığında, gömülecek damga verisi bir kez bu dosyadan okunur ve uygulama kapatılana kadar aynı veri ile tüm görüntüler damgalanır kullanılır. Bu şekilde damga verisi için dinamik bir durum sağlanmış olur. Gömü verisi değiştirilmek istendiğinde cihazdaki txt dosyasını değiştirmek yeterli olacaktır.



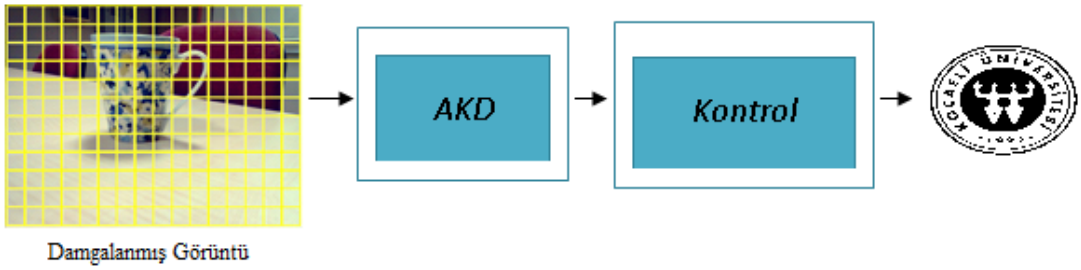
Şekil 2.5. Gömü verisi

Damga kontrolü için ise damgalama işlemi taklit edilir. Damga olduğu düşünülen görüntü 8x8 lik bloklara ayrılarak her bir bloğa AKD uygulanır. Her bloktaki (4,3) ve (5,2) nolu katsayılar kontrol edilir. (5,2) de bulunan katsayı (4,3) de bulunan katsayıdan büyük ise gömü verisi "1" değilse "0" kabul edilir. Damga kontrol işleminin ifadesi Denklem (2.9) daki gibi;

$$w = \begin{cases} 1, & (4,3) < (5,2) \\ 0, & (4,3) > (5,2) \end{cases} \tag{2.9}$$

formüle edilebilir.

Elde edilen 4800 elemanlı bit dizisinden, “1” ler siyah, “0” lar beyaz çizdirilerek 80x60 boyutlarında gömü verisinin görsel hali elde edilir (Şekil 2.6).



Şekil 2.6. Damga tespiti

3. BULGULAR

Bu bölümde geliştirilen uygulama aracılığı ile çekilen damgalanmış 6 görüntüye çeşitli saldırılar uygulanarak damganın dayanıklılığı test edilmiş ve açıkça sunulmuştur. Aynı damgalama metodu sabit görüntülere uygulanmış, farklı dayanıklılık sabiti değerleri için PSNR değerleri hesaplanmıştır. Aynı şekilde farklı dayanıklılık sabiti değerleri için damga çıkarım doğruluk oranı da hesaplanarak sunulmuştur.

Görüntü damgalama çalışmalarında dayanıklılığın yanında PSNR (Peak Signal to Noise Rate) da önemli bir değerlendirme kriteridir. PSNR, orijinal görüntüyü baz alarak, damgalanmış görüntüdeki bozulma değerini hesaplar (Ichigaya ve diğ., 2006). Bu tez çalışmasında geliştirilen uygulamada, orijinal görüntü hiçbir zaman var olmadığı için, telefon kamerasından alınan orijinal görüntüler ile damgalanmış arasında PSNR hesaplanamaz. Bu sorunu çözmek için, bu çalışmada kullanılan damgalama yöntemi sabit görüntülere uygulanmıştır. Lena, Baboon, Peppers, Tiffany görüntüleri damgalanarak PSNR değeri hesaplanmıştır. PSNR değeri hesaplanması için öncelikle MSE değeri hesaplanmalıdır. PSNR ve MSE değerleri Denklem (3.1) ve (3.2) deki gibi;

$$\mathrm{MSE} = \frac{\sum_{M,N}\left[\mathrm{I}(i,j) - \mathrm{K}(i,j)\right]^2}{\mathrm{M} \times \mathrm{N}} \tag{3.1}$$

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}}\right) \tag{3.2}$$

şeklinde hesaplanır.

Denklem (3.1) deki I ve K birbiriyle karşılaştırılacak görüntüleri temsil eder. I orijinal görüntü, K ise damgalanmış görüntüdür. Denklem (3.2) deki MAX değeri ise karşılaştırılan değerlerin maksimum alabileceği en büyük değerdir. Bir piksel 8 bitle ifade edildiğine göre maksimum değeri 255 olacaktır. Bu nedenle MAX=255'tir.

Renkli görüntüde bir piksel Kırmızı, Yeşil ve Mavi renklerinin birleşmesinden meydana geldiği için MSE hesaplanırken, orijinal ve yeniden elde edilmiş görüntülere ait piksellerin farklarının kareleri, görüntü boyutunun üç katına bölünür (URL-3).

Dayanıklılık katsayısı (α) değerini belirlemek amacıyla, görüntüler, dayanıklılık sabiti değeri 100-900 aralığında değiştirilerek damgalanmış, sonrasında damga çıkarım yüzdesi ve saldırılara dayanıklılığı test edilmiştir (Tablo 3.1). Dayanıklılık katsayısı α değeri düştükçe PSNR değeri olumlu yönde artmaktadır. Görüntüler farklı olmasına rağmen değerler birbirine oldukça yakındır. Dayanıklılık katsayısı α değeri değişimine karşı farklı görüntülerin PSNR değişim miktarı da birbirine yakın seyretmektedir. Ancak α değeri düştükçe damga çıkarım başarım yüzdesi ve saldırılara karşı dayanıklılığı düşmektedir. Farklı α değerleri için damgalanmış 4 adet görüntünün PSNR değerleri Tablo 3.1 de verilmiştir. Farklı α değerleri için örnek "Lena" görüntüsünden çıkarılan damga çıkarım yüzdeleri Tablo 3.2 de verilmiştir.

Tablo 3.1. Dayanıklılık sabiti(α) değişimi sonucu PSNR değerleri

	α=900	α=800	α=700	α=600	α=500	α=400	α=300
	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR
 Lena.jpg	36,14	36,68	37,23	37,81	38,37	38,93	39,43
 Baboon.jpg	34,01	34,39	34,76	35,13	35,48	35,79	39,43

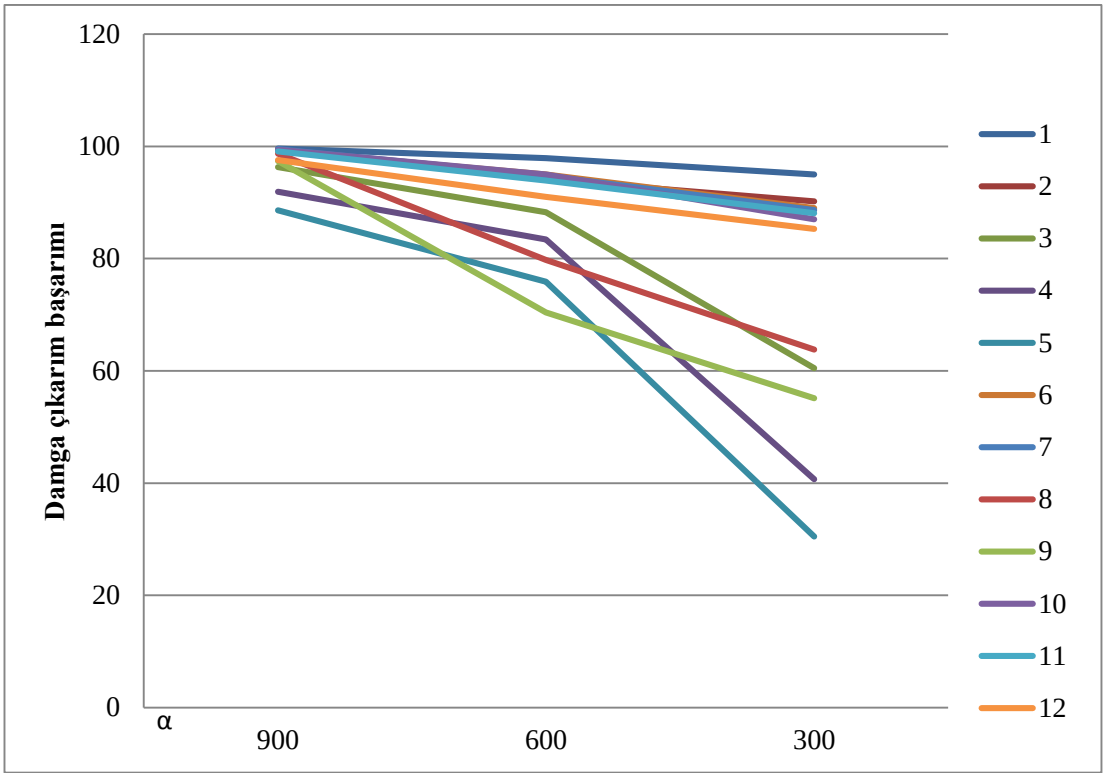
Tablo 3.1. Dayanıklılık sabiti(α) değişimi sonucu PSNR değerleri (Devam)

	α=900	α=800	α=700	α=600	α=500	α=400	α=300
	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR
 Peppers.jpg	34,70	35,19	35,70	36,23	36,76	37,27	37,75
 Tiffany.jpg	36,78	37,33	37,90	38,48	39,07	39,65	40,18

Tablo 3.2. Farklı α değerleri için damga çıkarımı

	α=900	α=800	α=700	α=600	α=500	α=400	α=300
 "Lena.jpg"	Çıkarılan damga: %98,85	Çıkarılan damga: %98,6	Çıkarılan damga: %98,3	Çıkarılan damga: %97,9	Çıkarılan damga: %97,37	Çıkarılan damga: %96,5	Çıkarılan damga: %95

Farklı dayanıklılık sabiti değerleri ile damgalanmış "Peppers.jpg" görüntüsünde damga kontrolü yapılmasının ardından, birbirinden bağımsız ve ayrı olarak 11 farklı saldırı yapılmış, sonrasında damga kontrolü yapılmıştır. Elde edilen sonuçlar Şekil 3.1 ve Tablo 3.3 de açıkça sunulmuştur.



Şekil 3.1. Damga Çıkarım Başarım Grafiği

Tablo 3.3. Dayanıklılık sabiti değişimi ve damga çıkarım başarımı

	1	2	3	4	5	6	7	8	9	10	11	12
α=900	%99,6	%99,4	%96,3	%91,9	%88,6	%99,2	%99,5	%98,7	%97,4	%99,4	%99,1	%97,6
α=600	%97,9	%94,1	%88,3	%83,4	%75,9	%95	%94,8	%79,8	%70,4	%95	%93,9	%91
α=300	%95	%90,2	%60,5	%40,7	%30,5	%89	%88,7	%63,8	%55,1	%87	%88,1	%85,3

1.Saldırı uygulanmadan damga çıkarım başarımı, 2. Descpeckle saldırısı, 3. Gaussian %3 saldırısı, 4. Gaussian %6 saldırısı, 5. Gaussian %10 saldırısı, 6. Sharpen saldırısı, 7. Sharpen Edge saldırısı, 8. Resize %80, 9. Resize %60, 10. Jpeg Quality=80, 11. Jpeg Quality=60, 12. Jpeg Quality=50

Dayanıklılık sabiti değeri 900 iken tüm saldırılara karşı yaklaşık %90 ve üzerinde damga çıkarım doğruluğu elde edilmiştir. Oldukça iyi olan bu sonuçlar, saldırıların damgayı çok az etkilediğini göstermektedir. Dayanıklılık sabiti değeri düştükçe ise damga çıkarım doğruluk oranın, bazı saldırılar sonrası %30 değerlerine düştüğü görülmektedir.

Geliştirilen uygulama aracılığı ile çekilen damgalı 6 görüntüye (Şekil 3.2) Photosop CS3 programı aracılığı ile Despeckle, Gaussian, Sharpen ve Sharpen Edge filitreleri ile yeniden boyutlandırma (resize) ve Jpeg sıkıştırma saldırıları uygulanmıştır.



Şekil 3.2. Çekilen damgalı görüntüler

Damgalanmış altı örnek görüntüden (Şekil 3.2) çıkarılan damgalar ve damgalı görüntülere yapılan saldırılar sonrasında çıkarılan damgalar Tablo 3.5 ve Tablo 3.6 da sunulmuştur. Uygulanan saldırılar sonucu görüntülerin son halini göstermek amacıyla da Şekil 3.2 de yer alan "b" görüntüsünün uygulanan saldırılar sonrası oluşan son görüntüleri Tablo 3.4'de örnek olarak sunulmuştur.

Tablo 3.4. Damgalı görüntüye çeşitli saldırıların uygulanması sonrası görüntüler

Despeckle	Gaussian %3
	
Gaussian %6	Gaussian %10
	
Sharpen	Sharpen Edge
	
Resize %80	Resize %60
	
Jpeg Quality=80	Jpeg Quality=60
	
Jpeg Quality=50	
	

Tablo 3.4 ve Tablo 3.5 deki (a),(b),(c),(d),(e),(f) sütunları Şekil 3.1'deki (a),(b),(c),(d),(e),(f) görüntülerini göstermektedir.

Yüzdeli ifadeler, çıkarılan damga ile orijinal damga kıyaslanarak elde edilen doğruluk yüzdeleridir. Gaussian ve yeniden boyutlandırma saldırılarından diğer saldırılara oranla daha yüksek oranda etkilendiği söylenebilir. Tablo 3.4'te görüldüğü gibi yapılan saldırılar sonrası görüntüde Gaussian saldırısı haricinde gözle görünür bir bozunma olmamıştır. Eğer damgalama yöntemi bu saldırılara karşı dayanıklı değilse veya bir diğer deyişle bu saldırılar sonrası damga verisi görüntü içerisinden silinirse, yetkisiz kişiler tarafından istenildiği gibi kullanılabilir.

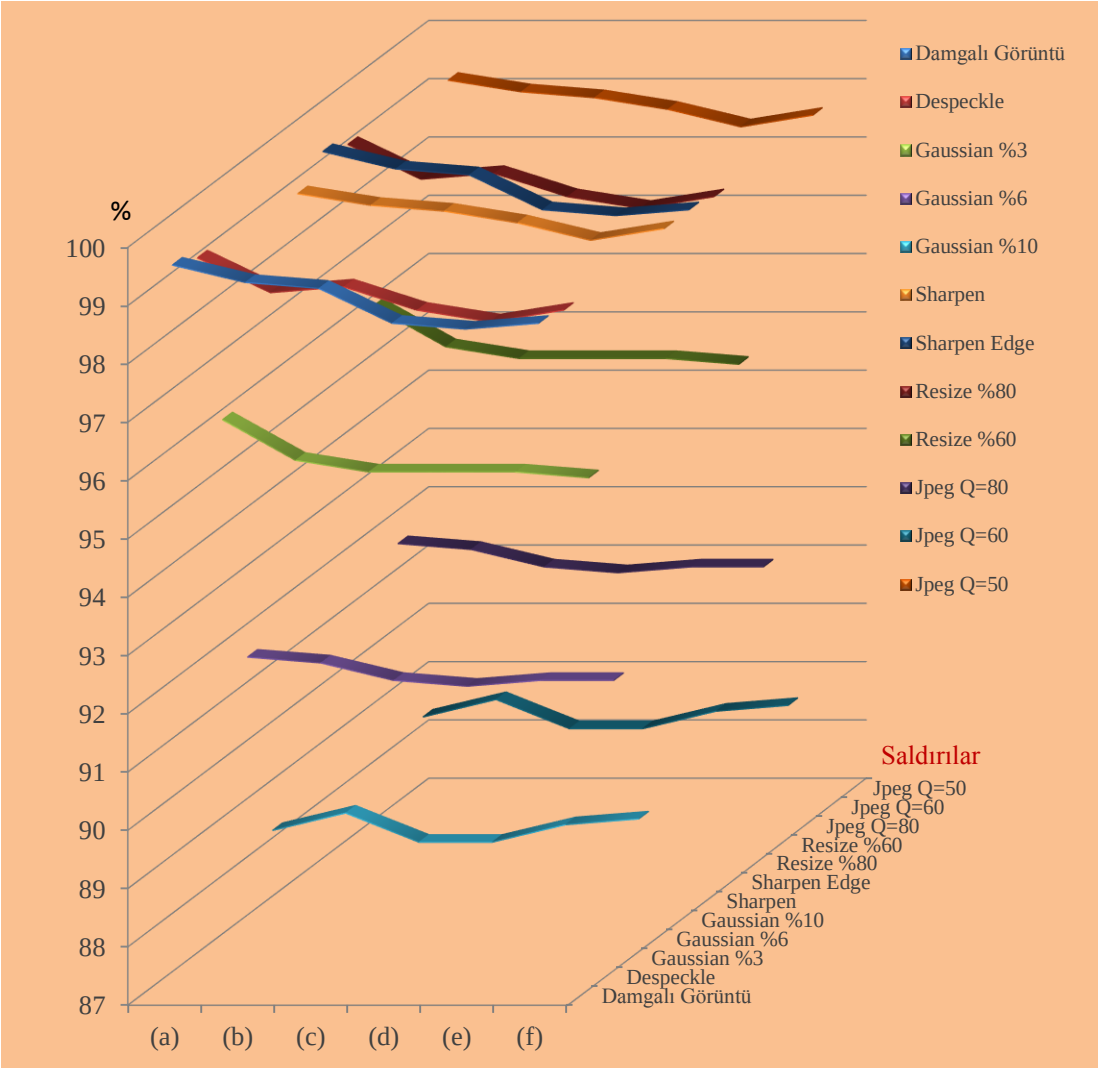
Tablo 3.5. Damga doğrulukları 1

	Damga Çıkarımı	Despeckle	Gaussian %3	Gaussian %6	Gaussian %10	Sharpen
(a)	%99.6	%99,4	%96,3	%91,9	%88,6	%99,2
(b)	%99,3	%98,8	%95,6	%91,8	%88,9	%99
(c)	%99,2	%98,9	%95,4	%91,5	%88,4	%98,9
(d)	%98,6	%98,5	%95,4	%91,4	%88,4	%98,7
(e)	%98,5	%98,3	%95,4	%91,5	%88,7	%98,4
(f)	%98,6	%98,5	%95,3	%91,5	%88,8	%98,6

Fotoğraf çekme sırasında ekranda görüntülenen damgalı her bir frame Bitmap formatındadır. Fakat fotoğrafın Nokia Lumia 820'nin galerisine (Media Library) kaydedilebilmesi işlemi ancak JPEG formatında yapılabilmektedir. Bu nedenle kaydedilen her bir görüntü cihaz tarafından JPEG sıkıştırma saldırısına maruz kalır.

Tablo 3.6. Damga doğrulukları 2

	Sharpen Edge	Resize %80	Resize %60	Jpeg Quality=80	Jpeg Quality=60	Jpeg Quality=50
(a)	%99,5 	%98,7 	%97,4 	%99,4 	%99,1 	%97,6 
(b)	%99,2 	%97,6 	%96,1 	%99,1 	%98,4 	%97,2 
(c)	%99,1 	%98,1 	%96,8 	%99,1 	%98,9 	%97,2 
(d)	%98,8 	%97 	%95,5 	%98,7 	%97,9 	%97,0 
(e)	%98,4 	%97,3 	%96,0 	%98,6 	%98,1 	%96,6 
(f)	%98,5 	%97,7 	%96,4 	%98,9 	%98,5 	%96,8 



Şekil 3.3. Damga çıkarım doğruluk grafiği

4. SONUÇLAR VE ÖNERİLER

Bu çalışmada Windows Phone 8 platformu üzerinde damgalayarak fotoğraf çeken bir mobil uygulama tasarlanmıştır. Gelişen teknolojinin sağlamış olduğu pek çok yenilik, uzun süredir gerçekleştirilen ve iyi bilinen bir konu olan dayanıklı görüntü damgalama konusu ile birlikte ele alınmıştır. Platformun sağladığı Native Code ile çalışma zamanında (runtime) kameranın yakaladığı görüntülere erişebilme özelliği ve görüntü verilerine hafıza üzerinden doğrudan erişebilme imkânı ile oldukça performanslı bir uygulama geliştirilmiştir. Uygulama, görüntüleri kameradan alıp oldukça yüklü işlemler gerektiren, dayanıklı bir metot olan AKD temelli yöntem ile damgalayarak, ekranda saniyede 640x480 boyutlarında 4-5 görüntü karesi gösterebilmesi yönünden, bu performanstaki ilk mobil çalışma olarak değerlendirilmektedir.

Damgalanmış görüntülere pek çok saldırı uygulanarak test edilmiş, çoğu saldırıya karşı dayanıklı olduğu tespit edilmiştir. Ayrıca facebook, instagram ve whats up gibi görüntü paylaşımının sık kullanıldığı sosyal medya ortamlarına da damgalanmış görüntüler yüklenerek test edilmiş ve damganın korunduğu görülmüştür. PSNR değerini test etmek amacı ile damgalama metodu farklı sabit görüntülere uygulanmış ve damgalanmış görüntü ile orijinal görüntü arasındaki bozunma, PSNR>30 olduğu için, görüntüdeki bozulma kabul edilebilir seviyededir.

Her geçen gün daha hızlı ve daha performanslı akıllı telefonlar piyasaya sürülmektedir. Sonraki çalışmalarda daha güçlü işlemciye sahip daha hızlı akıllı telefonlar üzerinde, hızlı AKD algoritmaları (Sookim ve Teapark, 1998) ile 30 fps hızında eşzamanlı görüntü damgalama uygulamaları gerçekleştirilebilir. Görüntü bloklara ayrılıp, her bir bloğa birbirinden bağımsız işlemler uygulandığından, çok çekirdekli mobil platformlarda işlemler parçalara ayrılarak ve farklı çekirdeklere paylaştırılarak daha verimli uygulamalar geliştirilebilir.

KAYNAKLAR

Akleylek S., Nuriyev U., Steganography and new implementation of steganography, *IEEE 13th Signal Processing and Communications Applications Proceedings Conference*, Kayseri, Turkey, 13-14 May 2005.

Barni M., Bartolini F., Cappellini V., Lippi A., Piva A., DWT-based technique for spatio-frequency masking of digital signatures, *International Conference Society for Optics and Photonics*, San Jose, California, 23 January 1999.

Cedillo-Hernandez M., Garcia-Ugalde F.J., Nakano-Miyatake M., Perez-Meana H.M., DFT-based watermarking method for medical images, *IEEE International Ninth Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, Mexico, 19-23 November 2012a.

Cedillo-Hernandez M., Garcia-Ugalde F.J., Nakano-Miyatake M., Perez-Meana H.M., Robust digital image watermarking using interest points and DFT domain, *IEEE 35th International Telecommunications and Signal Processing Conference*, Prague, Czech, 3-4 July 2012b.

Chen Y. Y., Medical image compression using DCT-based subband decomposition and modified SPIHT data organization, *International journal of medical informatics*, 2007, **76**, 717-725.

Chen Y. H., Huang H. C., A copyright information embedding system for android platform, *Seventh International Intelligent Information Hiding and Multimedia Signal Processing Conference*, Dalian, China, 14-16 October 2011.

Cummins J., Diskin P., Lau S., Parlett R., Steganography and Digital Watermarking, The University of Birmingham, http://www.cs.bham.ac.uk/~mdr/teaching/ modules03/security/students/SS5/Steganography.pdf, (Ziyaret tarihi: 20 Ağustos 2014).

Çetin Ö., Özcerit A. T., A new steganography algorithm based on color histograms for data embedding into raw video streams, *Computers & Security*, 2009, **28**, 670-682.

Huang J., Shi Y. Q., Shi Y., Embedding image watermarks in DC components, *Circuits and Systems for Video Technology*, 2000, **10**, 974-979.

Ichigaya A., Kurozumi M., Hara N., Nishida Y., Nakasu E., A method of estimating coding PSNR using quantized DCT coefficients, *Circuits and Systems for Video Technology*, 2006, **16**, 251-259.

Jain A. K., Image transforms, Editor: Thomas Kailath, *Fundamentals of digital image processing*, 2nd ed., Prentice Hall, New Jersey, 150-153, 1989.

Jassim T., Abd-Alhameed R., Al-Ahmad H., A new robust and fragile watermarking scheme for images captured by mobile phone cameras. *In Communications, Signal Processing, and their Applications* , Sharjah, 12-14 February 2013.

Jingbing L., Yaoli L., Wencai D., Yen-Wei C., The medical image watermarking algorithm based on DFT and logistic map, *7th International Computing and Convergence Technology Conference*, Seoul, 3-5 December 2012.

Ketcham M., Vongpradhip S., Intelligent audio watermarking using genetic algorithm in DWT domain, *International Journal Of Intelligent Technology*, 2007, **2**, 135-140.

Koch E., Zhao J., Towards robust and hidden ımage copyright labeling, *IEEE Workshop on Nonlinear Signal and Image Processing*, Neos Marmaras, Greece, 20-22 June1995.

Kundur D., Hatzinakos D., A robust digital ımage watermarking scheme using wavelet-based fusion, *IEEE Int. Conf. Image Processing*, Santa Barbara, California, 26-29 October 1997.

Lee Y. K., Chen L. H., High capacity image steganographic model, *IEEE Vision, Image and Signal Processing*, 2000 , **147**, 288-294.

Ling L., Xinde S., Leiting C., A robust image watermarking based on DCT by arnold transform and spread spectrum, *3rd International Advanced Computer Theory and Engineering Conference*, Chengdu, China, 20-22 August 2010.

Mehta S., Nallusamy R., Marawar R. V., Prabhakaran B., A study of DWT and SVD based watermarking algorithms for patient privacy in medical images, *IEEE International Healthcare Informatics Conference*, Philadelphia, United States, 9-11 September 2013.

Miao N., He Y., Dong J., HymnMark: Towards Efficient Digital Watermarking on Android Smartphones, Worldcomp-Proceedings, http://worldcomp-proceedings.com/proc/p2012/ ICW6177.pdf, (Ziyaret tarihi: 30 Haziran 2014).

Palma Hernandez C., Torres-Huitzi C., A fragile watermarking scheme for image authentication in mobile devices, *International Electrical Engineering Computing Science and Automatic Control Conference*, Merida , Mexico, 26-28 October 2011.

Patil P. M., Shinde S., Arekar C. V., Bhandwalkar N. M., Copyright for images with chaotic sequence, *International Journal of Research*, 2014, **1**, 96-99.

Shahreza S., Stealth steganography in SMS, *IFIP International Wireless and Optical Communications Networks Conference*, Bangalore, India, 11-13 April 2006.

Simmons G. J., The prisoners' problem and the subliminal channel, Editor: David Chaum, *Advances in Cryptology* , 1nd ed., Springer US, Albuquerque, 51-67, 1984.

SooKim H., TaePark K., Fast DCT algorithm with fewer multiplication stages, *Electronics Letters*, 1998, **34,** 723-724.

URL-1: http://www.whydomath.org/node/wavlets/jpeg2000quantization.html, (Ziyaret Tarihi: 20 Ağustos 2014).

URL-2: http://msdn.microsoft.com/en/us/library/windowsphone/develop/jj681687 (v=vs.105).aspx, (Ziyaret tarihi: 30 Haziran 2014).

URL-3: http://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio#searchInput#search Input, (Ziyaret tarihi: 30 Haziran 2014).

Wang B., Ding J., Wen Q., Liao X., Liu C., An image watermarking algorithm based on DWT DCT and SVD, *International Network Infrastructure and Digital Content Conference*, Beijing, China, 6-8 November 2009.

Yeşilyurt M., Özcerit A. T., Yalman Y., Ertürk İ., Sayısal imgeler için ayrık kosinüs dönüşümü esaslı veri gizlemenin ataklara karşı dayanıklılığı, *Akademik Bilişim Konferansı*, Uşak, Türkiye, 1-3 Şubat 2012.

KİŞİSEL YAYIN VE ESERLER

Uçar A., Becerikli Y., Mobil Ortamda AKD Tabanlı Görüntü Damgalama Uygulaması, *7th International Conference on Information Security and Cryptology*, İstanbul, Türkiye, 17-18 Ekim 2014.

Uçar A., Topel A.A., Onay Durdu P., Fatih Projesi Etkileşimli Tahta Kullanılabilirlik Sorunları ve Teknoloji Kabul Düzeyi Üzerine Yapılan Pilot Çalışma, *8. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, Edirne, Türkiye, 18-20 Eylül 2014.

Uçar A., Onay Durdu P., Akıllı tahta kullanılabilirlik çalışması, *7. Uluslararası Bilgisayar ve Öğretim Teknolojileri Sempozyumu*, Erzurum, Türkiye, 6-8 Haziran 2013.

ÖZGEÇMİŞ

19.06.1988 tarihinde Malatya'da doğan Ali UÇAR, 2005 yılında Alanya Lisesinden mezun oldu ve aynı yıl Süleyman Demirel Üniversitesi Teknik Eğitim Fakültesi, Elektronik Bilgisayar Eğitimi Bölümünde lisans öğrenimine başladı. Bir yılı ingilizce hazırlık olmak üzere beş yıllık eğitimin ardından 2010 yılında mezun oldu. Aynı yıl Milli Eğitim Bakanlığı tarafından öğretmen olarak atandı ve 3 yıl süresinde öğretmenlik, MEBBİS yöneticiliği, Fatih projesi eğitmeni pozisyonlarında çalıştı. 2010-2011 eğitim öğretim yılı, bahar yarıyılında Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalında yüksek lisans eğitimine başladı ve 2013 yılından beri aynı bölümde Araştırma Görevlisi olarak çalışmaktadır.