Çankaya University Journal of Science and Engineering
Volume 10 (2013), No. 1, 17–32

İmge İçine Bilgi Gizlemede Kullanılan LSB Yöntemlerinin Karşılaştırması

Cem Olcay[1] ve Nurdan Saran[2,*]

[1] *Fen Bilimleri Enstitüsü, Yaşar Üniversitesi, 35100, İzmir, Türkiye*
[2] *Çankaya Üniversitesi, Bilgisayar Mühendisliği Bölümü, 06810, Yenimahalle, Ankara, Türkiye*
[*] *Corresponding author: buz@cankaya.edu.tr*

Özet. Dijital imgeleri temel olarak kayıplı sıkıştırılmış tipte olanlar, sıkıştırılmamışlar ve kayıpsız sıkıştırılmışlar olarak üçe ayırabiliriz. Sıkıştırılmamış ve kayıpsız sıkıştırılmış imgelerde bilgi gizlemek için en çok kullanılan yöntem, imgenin son bitlerinin mesajın bitleri ile değiştirilmesi yöntemidir. Son bitlerin değişimi genellikle imgede gözle görülebilir bir değişikliğe sebep olmamaktadır. Örtü imgedeki piksellerin son bitlerinin değişimi ile veri gizleme yöntemlerini her renk kanalında 1 bit yada daha fazla veri gizleyenler ve birden çok renk kanalında renk kanalı sayısından daha az miktarda veri gizleyenler olarak ikiye ayırabiliriz. Bu çalışmada en önemsiz bite (least significant bit-LSB) gizleme yöntemlerinden yer değiştirme [1], eşleştirme [2], eşleştirmenin geliştirilmiş bir hali olan Chan'ın algoritması [3], 2/3 verimli gömme [4], Hamming kodlarını kullanılarak matris gömme [5], imge kareleri [6] ve piksel farkı [7]- [8] yöntemleri karşılaştırılmıştır.

Anahtar Kelimeler. En önemsiz bite gizleme, steganography, bilgi gizleme, imge içine bilgi gizleme.

Abstract. Digital image steganography techniques deal with three subjects lossy/lossless compressed images, uncompressed images. Steganography techniques embed secret data pixel directly into uncompressed and lossless compressed cover images. The most popular data hiding method is, changing pixels' left most digits or last two, known as Least Significant Bit (LSB). Lossless image formats like .bmp, .png, and 8-bit gray-scale .gif are usable for LSB methods. LSB embedding method is not usable for palette images because of changing just last bit of a pixel causes a big difference on image. After embedding all of secret message to palette image, Human Visual System (HVS) can detect manipulations on image. In this study, we compare some of the most popular LSB techniques; Chan's algorithm [3], 2/3 efficient embedding [4], matrix embedding for large payloads [5], image blocks method [6] and pixel-value differencing method [7]-[8].

Keywords. LSB technique, steganography, information hiding, digital image steganography.

Received April 5, 2012; accepted February 26, 2013.

ISSN 1309 - 6788

1. Giriş

Günümüzde bilgi teknolojilerinin geldiği nokta ile bilgi paylaşımının hız kazandığı ve kolaylaştığı ortadır. Fakat bu durum beraberinde bilgi güvenliği ile ilgili sorunlar da oluşturmaktadır. Gerek kişisel bilgilerin saklanması gerekse haberleşmede bilgi gizliliği için güvenlik önlemleri almanın önemi artmıştır. Dijital ortamda ilettiğiniz bütün bilgilerin takip edilebileceği düşünüldüğünde, bilgilerinizin ve mesajlarınızın şifreleme yöntemleri kullanılarak güvenliğini sağlamak bir önlem olabileceği gibi, şifreli bilgilerinde dikkat çekeceği aşikardır. Bu yüzden bilgi gizlemede steganografi teknikleri oldukça önemli yer tutmaktadır.

Steganografi bilgi gizleme yöntemlerinin bir alt dalıdır. Verinin gizlendiği ortama örtü ortamı (cover-media), oluşan ortama da stego-nesnesi (stego-object) denilmektedir. Genel olarak steganografi, tutuklu problemi (prisoner problem) ile açıklanabilir. A ve B, hapishaneden kaçmak için iletişime girecek iki kişi olsun, W de gardiyan olsun. A, X örtü ortamında gizli bilgi m'yi k gizli anahtarını kullanarak gömer. Karşı tarafta B, k gizli anahtarı ile m verisini X örtü ortamından geri dönüştürür. Steganografi, W'nin X'i ele geçirmesi durumunda bir bilgi gizlendiğini fark etmemesini sağlamayı amaçlar. Başka bir ifadeyle, steganografide amaç bir bilginin bir taşıyıcı kullanarak alıcıya iletimi sırasında taşıyıcının başka kişilerin eline geçmesi durumunda bir bilgi gizlendiğini farketmemesini sağlamaktır [9, 13].

Güvenlik, kapasite ve anlaşılmazlık bir steganografik tekniğin sağlaması gereken üç önemli gereksinim olarak ifade edilebilir.

- Güvenlik: Eğer mesajın varlığı rastgele tahminden daha yüksek bir olasılıkla tahmin edilemiyorsa, böyle bir yöntem steganografik olarak güvenlidir denilebilir.
- Kapasitesi: Steganografik tekniğin, örtü ortamına en fazla veriyi saklamaya çalışmasıdır, veri gizleme kapasitesi, α, gizli mesajın örtü imgeye oranı ile hesaplanır.
- Anlaşılmazlık: Stego nesnesinde veri gizlendiği farkedilecek kadar çok yapaylık (artifact) olmamalıdır. Bir steganografik teknik diğer tekniklerle aynı düzeyde kapasite ve güvenlik ölçülerine sahip iken, stego nesne ile orjinal nesne arasında anlaşılmazlık daha yüksekse (ϵ, değişen bitlerin oranı daha düşükse) o yöntemin daha iyi olduğu söylenebilir.

Örtü nesne olarak en çok kullanılan dijital medya imgelerdir. İmgeler frekans alanında ve uzamsal (spatial) etki alanında olanlar olarak ikiye ayrılırlar. Veri gömme

işlemlerinde frekans alanındakilerde kodlama ve sıkıştırma yöntemleri uygulandığı için piksel değerleri doğrudan kullanılamaz. Uzamsal alanda olanlar imgedeki verileri doğrudan kullandığı için imgedeki piksel değerlerinde yapılacak değişiklikler başka bir değişime uğramazlar.

Bir steganografik yöntemin başarısı, steganaliz yöntemlerine karşı güçlülüğüyle de değerlendirilmektedir. Steganalizin öncelikli amacı bir örtü nesnesinde bir bilginin gizli olup olmadığını tesbit etmektir. Nesnede bilgi gizlendiği fark edildiğinde ise gizli bilgiyi elde etmeyi hedefler. Gizli bilginin tesbitinin zor olması için stego nesne üzerinde yapılacak değişikliklerin en az seviyede olması gerekir. Bu nedenle, örtü nesne olarak bir imgenin kullanıldığı steganografik yöntemin başarısı hakkında konuşmak için MSE (ortalama karesel hata) ve PSNR (en üst sinyal gürültü oranı) değerlerine bakılmaktadır.

Ortalama karesel hata (MSE), iki imge arasındaki farkı belirtmek için piksel değerlerinin farklarının karelerinin tüm piksel değerine bölünmesidir. Eğer iki imge aynı ise MSE değeri 0'dır.

$$\mathrm{MSE} = \frac{\sum_{M,N} \left[I_1(m,n) - I_2(m,n) \right]^2}{M * N}$$

Yukarıdaki formülde I_1 ve I_2 aynı $M \times N$ boyutundaki imgelerdir.

En üst sinyal gürültü oranı (PSNR) değeri imgedeki değişiklikleri aşağıdaki formülle hesaplar. Eğer iki imge aynı ise PSNR sonsuz olur.

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

Yukarıdaki formülde R imgedeki en büyük piksel değeridir. Sekiz bitlik bir imge için $R = 2^8 - 1 = 255$ dir.

Bu makalede imgeler üzerinde kullanılan LSB yöntemlerinin herbir renk kanalında bir bit bilgi gizleyen ve en çok kullanılanlarından olan LSB yer değiştirme ± 1 yöntemi, LSB eşleştirme yöntemi, piksel farkı yöntemi ve LSB eşleştirmenin geliştirilmiş bir hali olan Chan'ın algoritması incelenmiştir. Ayrıca birden çok renk kanalında, renk kanal uzunluğundan daha az uzunlukta bilgi gizleyerek oluşturulacak imgede daha az değişikliği sağlamayı amaçlayan verimli gizleme tekniği, imge kareleri yöntemi ve matrisler halinde gizlemede Hamming kodunu kullanan yöntemleri incelenmiştir.

2. LSB Yöntemleri

LSB yöntemlerde örtü imgenin piksellerinin en önemsiz bitiyle gizli mesajın bitleri yer değiştirir. En basit şekliyle örtü imgenin satır veya sütünlarına sırayla gizli mesajın bitleri gizlenebilir ancak bu durumda mesajın geri elde edilmesi çok kolaylaşacaktır. LSB yöntemlerinde hangi piksellerde değişiklik olacağı bir çok farklı tekniklerle belirlenebilir, literatürde bir çok teknik önerilmiştir. Bu tekniklerin başlıcaları, ayrık logaritma fonksiyonu [14] ve Laplacian kenar bulma algoritmasıdır [15]-[16]. Bu yöntemler kullanılarak değiştirilecek pikseller rastgele belirlenebilir.

İmgeler farklı formatlarda olabileceğinden gizleme kapasiteleri de değişmektedir. Bu çalışmada incelenen bütün yöntemlerde sekiz bitlik gri tonlamalı imgeler kullanılmıştır. Sekiz bitlik gri tonlamalı imgelerde her bir piksel sekiz bit boyutunda tek renk kanalından oluşur.

Aşağıdaki yöntemlerde piksellerin en önemsiz biti olarak tanımlanan son bite göre bilgi gizlenmiştir. Makalenin bütünlüğü için açıklanan yöntemlerdeki notasyonlar şu şekilde olacaktır:

- örtü imgedeki i. pozisyondaki pikselin değeri y_i
- herhangi bir p_i pikselinin en önemsiz biti $p_{i,1}$
- herhangi bir p_i pikselinin en önemsiz 2. biti $p_{i,2}$
- gizlenecek verideki i. pozisyondaki verinin değeri m_i, m_i verisinin en önemsiz biti $m_{i,1}$
- stego imgedeki i. pozisyondaki pikselin değeri ise $\hat{y}_i$

Seçilen yönteme göre imgede kullanılan piksellerde gizlenen bit oranı α, bu piksellerden değişime uğrayanların oranı ϵ ile gösterilmiştir.

2.1. Yer değiştirme yöntemi ± 1. Bu yöntemde imgedeki her bir pikselin en değersiz biti gizlenecek bilginin o piksele denk gelen biti ile karşılaştırılarak bir bit veri gizlenir ($\alpha = 1/1$). Örneğin, piksel değeri 6 ise bunun ikilik sistemde yazımı 00000110'dır. Eğer bu pikselde 1 gizlenecekse, pikselin değeri bir arttırılır yada azaltılır, eğer 0 gizlenecekse herhangi bir değişiklik yapılmaz. Bu yöntemde sekiz bitte gizlenen sekiz bit veri için dört bitte değişiklik oluşmaktadır. İmgede bilgi gizlenen piksellerin ortalama %50 sinde değişim oluşmaktadır ($\epsilon = 2/1$). Aşağıdaki Tablo 1'de örtü piksel bitleri ikili olarak gruplanmıştır.

Tablo 1. Yer değiştirme yöntemiyle bilgi gizleme.

$m_{i,1}m_{i,2}$ \ y_iy_{i+1}	6 6	7 7	6 7	7 6
0 0	6 6	6 6 veya 8 8	6 6 veya 6 8	6 6 veya 8 6
0 1	6 7 veya 6 5	6 7 veya 8 7	6 7	6 7 veya 8 5
1 0	7 6 veya 5 6	7 6 veya 7 8	7 6 veya 5 8	7 6
1 1	7 7 veya 5 5	7 7	7 7 veya 5 7	7 7 veya 7 5

2.2. Eşleştirme yöntemi. Bu yöntemde amaç örtü imgede daha az değişikliği sağlayarak her bir piksele bir bit gizlemektir ($\alpha = 1/1$). Örtü verideki pikseller ve gizlenecek veriler ikili gruplar halinde değerlendirilerek gizlenir. Mielikainen [2] tarafından 2006'da önerilen bu yöntemin çalışma şekli aşağıda verilmiş ve Tablo 2'de örnek durumlar belirtilmiştir.

1. Örtü imgedeki iki pikselin degerleri y_1, y_2, gizlenecek mesaj m_i olsun.
2. $y_{1,1}$ ile $m_{i,1}$ karşılaştırılır.
3. Eğer eşitlerse $\hat{y}_1$, y_1'e eşit olarak tanımlanır.
   (a) $y_{1,2}$ ve $y_{2,1}$'nin toplamının son biti, $m_{i,2}$ ile karşılaştırılır.
      (i) Eğer eşitlerse $\hat{y}_2$, y_2'ye eşit olarak tanımlanır.
      (ii) Eğer eşit değillerse y_2, 1 arttırılır ya da azaltılır.
4. Eğer eşit değillerse $\hat{y}_2$, y_2'ye eşit olarak tanımlanır.
   (a) $y_{1,2} - 1$ ve $y_{2,1}$'nin toplamının son biti, $m_{i,2}$ ile karşılaştırılır.
      (i) Eğer eşitlerse $\hat{y}_1$'in değeri $y_1 - 1$'dir.
      (ii) Eğer eşit değillerse $\hat{y}_1$'in değeri $y_1 + 1$'dir.

Tablo 2. Mielikainen'in Eşleştirme yöntemiyle bilgi gizleme.

$m_{i,1}m_{i,2}$ \ y_iy_{i+1}	6 6	7 7	6 7	7 6
0 0	6 7	6 7	6 7	8 6
0 1	6 6	8 7	6 6	6 6
1 0	5 6	7 7	7 7	7 7
1 1	7 6	7 6	5 7	7 6

Bu yöntemde sekiz bit gizlendiği durumda üç bitte değişiklik olmaktadır ($\alpha = 8/3$). Yer değiştirme yöntemine göre %12.5'luk kapasite avantajına sahiptir.

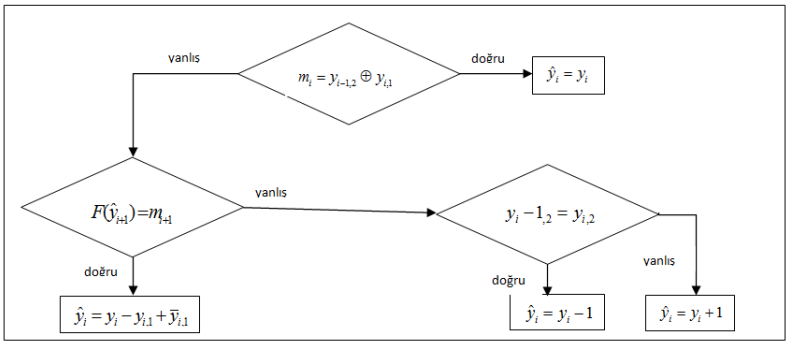
2.3. Chan'ın eşleştime yöntemi. Eşleştirme yöntemine göre aynı kapasitede bilgi gizlerken ($\alpha = 1/1$) daha az değişlik oluşturmayı amaçlayan bu yöntemde bir piksele değer gizlenirken kendisinden önceki değerle karşılaştırılır, eğer kullanılan formüle uygunsa değişiklik yapılmaz [3]. Formüle uygun değilse kendinden sonra gelen piksel ile karşılaştırılıp duruma göre piksel değeri değiştirilir ve bilgi gizlenir.

Chan, Mielikainen'in eşleştirme yönteminden esinlenerek değiştirilen piksel sayısının azaltılması için bu yöntemi önermiştir. Bu yöntemle her pikselde bir bit bilgi gizlenerek diğer yöntemlere göre imgede daha az değişiklik oluşturulması amaçlanmıştır. Mielikainen'in yönteminde örtü imgedeki pikseller ile gizlenecek bitler ikili gruplar halinde kodlanırken, Chan'ın yönteminde gizlenecek bitler örtü imgeye sıralı olarak yerleştirilir. Diğer yöntemlerde imgedeki değişim oranı gizlenen bilginin uzunluğuyla bağlantısı olmazken Chan'ın geliştirdiği yöntemde gizlenen bilginin artmasıyla imgedeki değişim oranı düşmektedir.

Chan'ın yönteminin çalışma şekli aşağıda açıklanmış ve işleyiş diyagramı Şekil 1'de verilmiştir.

1. Öncelikle imgenin i. pozisyondaki piksel değerinin son biti ile $i-1$ pozisyondaki pikselin sondan 2. biti XORlanır ve i. pozisyonda gizlenecek bit ile karşılaştırılır.
2. Eğer bu karşılaştırmanın sonucunda 2 değer eşitse stego imge için o pikselde bir değişiklik yapılmaz.
3. Eğer karşılaştırmanın sonucunda 2 değer eşit değilse F fonksiyonu kullanılır. F fonksiyonunda i. pozisyondaki pikselin sondan 2. biti ile $i+1$. pozisyondaki pikselin son biti XORlanır. Çıkan sonucun $i+1$ pozisyonunda gizlenecek bit ile eşitliği karşılaştırılır.
4. Eğer eşit değillerse, i. pozisyondaki pikselin sondan 2. biti ile o pozisyondaki pikselin 1 eksik değerinin sondan 2. biti karşılaştırılır.
5. Eğer bu 2 değer eşitlerse stego imge için o pozisyondaki değer 1 azaltılır.
6. Eğer bu 2 değer eşit değillerse stego imge için o pozisyondaki değer 1 arttırılır.
7. Eğer 3. adımdaki karşılaştırmanın sonucu eşitse stego imgede o pozisyonun değerini bulmak için pikselin değerinden son bitteki değeri çıkartılır ve son bitteki değerinin tümleyeni eklenir.

Örnek olarak örtü imgedeki piksel değerinin 2 olduğunu ve kendinden önceki pikselin LSB değerinin 0 olduğu varsalım.



ŞEKİL 1. Chan'ın yönteminin isleyiş diyagramı.

- O piksele gizli bit olarak 0 gizlenmek istenirse;

$y_{i-1,1}$	$y_{i,2}y_{i,1}$		$m_{i,1}$
0	0	1	0

  Sonuç: birinci durum, değişikliğe gerek yok.
- Eğer örtü imgedeki arda arda gelen 2 pikselin değeri 2,2 ve bu piksellerde gizlenecek bitler 1,1 ise;

$y_{i-2,1}$	$y_{i-1,2}y_{i-1,1}$		$y_{i,2}y_{i,1}$		$m_{i,2}m_{i,1}$	
0	1	0	1	0	1	1

  Sonuç: Aynı renkli bitlerin XOR sonucu; 0 1, gizlenecek bitler 11'e uyuşmadığından ikinci ve altıncı durum stego imgedeki o pozisyondaki değer aşağıdaki tabloda verildiği gibi bir artırılır.

$y_{i-2,1}$	$y_{i-1,2}y_{i-1,1}$		$y_{i,2}y_{i,1}$		$m_{i,2}m_{i,1}$	
0	1	1	1	0	1	1

  Sonuç: Aynı renkli bitlerin XOR sonucu; 1 1, gizlenecek bitler 11'e eşit.
- Eğer örtü imgedeki arda arda gelen iki pikselin değeri 2,2 ve bu piksellerde gizlenecek bitler 10 ise;

$y_{i-2,1}$	$y_{i-1,2}y_{i-1,1}$		$y_{i,2}y_{i,1}$		$m_{i,2}m_{i,1}$	
0	1	0	1	0	1	0

  Sonuç: Aynı renkli bitlerin XOR sonucu; 0 1 , gizlenecek bitler 10 la uyuşmadığından dördüncü ve beşinci durum stego imgedeki o pozisyondaki değer aşağıdaki tabloda verildiği gibi bir azaltılır.

$y_{i-2,1}$	$y_{i-1,2}y_{i-1,1}$		$y_{i,2}y_{i,1}$		$m_{i,2}m_{i,1}$	
0	0	1	1	0	1	0

  Sonuç: Aynı renkli bitlerin XOR sonucu; 1 0, gizlenecek bitler 10'a eşit.

Birinci durumda mevcut pozisyonda bit gizlemek için değişliklik gerekmediğinden değişiklik yapılmamıştır. Üçüncü durumda mevcut pozisyondaki durum için değişiklik gerekmektedir. Bu değişikliği yapmak için kendisinden sonraki pikselde gizlenecek bit için değişiklik gerekip gerekmediği kontrol edilmiştir. Gerekmediği sonucuna varıldığında sadece mevcut pozisyondaki biti gizleyebilmek amaçlı değişiklik yapılmıştır. İkinci örnekte ise mevcut pozisyondaki pikselde ve bir sonraki pikselde de değişiklik gerektiğinden, yapılan değişikle bir sonraki pozisyona bit gizlenirken bir değişiklik yapılmaması sağlanmıştır.

Gizli bilginin fazla olması bu yöntem için bir avantaj olduğundan yukarıda bahsedilen yöntemlerde kullanılan örnek sekiz piksele uygulanmış ve sonuçları Tablo 3'te verilmiştir.

Tablo 3. Chan'ın yöntemi ile bilgi gizleme.

Örtü imgedeki pikseller	66776776	66776776	66776776	66776776
Gizlenecek bitler	00000000	01010101	10101010	11111111
Stego imgedeki değerleri	67777777	66786676	56676786	76876876

2.4. 2/3 Verimli gizleme yöntemi. Bu yöntemle iki bit bilgiyi gizlemek için üç piksel kullanılır ve bu piksellerin değerinin değişmesi gerekirse 1 arttırılır veya 1 azaltılır [17]. Gizlenecek iki bit dışında kalan bitin değeri diğer iki bitin aynı şekilde gizlendiğini veya tümleyenlerinin gizlendiğini belirtir. Aşağıdaki Tablo 4'te ikili gizlenecek bit gruplarının bu yöntemle üçlü bit grupları halinde kodlanırken aldıkları değerler belirtilmiştir ve farklı bit gruplarında bilgi gizlenirken oluşabilecek değişiklikler Tablo 5'te gösterilmiştir.

Tablo 4. 2/3 Verimli gizleme yöntemiyle kodlama.

$m_{i,2}m_{i,1}$	00	01	10	11
$\hat{y}_{i,1}\hat{y}_{i+1,1}\hat{y}_{i+2,1}$	000 111	001 110	010 101	011 100

Bu yöntemin bilgi gizleme kapasitesi $\alpha = 2/3$ ve Gizlenen Bit/Değişen Bit oranı $\epsilon = 3/8$' dir. Bu yöntemin imgede yarattığı değişim oranı Mielikainen'in yöntemiyle aynı oranda olmasında karşın %33'lük daha az bilgi gizleme kapasitesine sahip olduğu için bir dezavantaja sahiptir.

TABLO 5. 2/3 Verimli gizleme yöntemiyle değişik piksellere bilgi gizleme örneği ($m_i m_{i+1}$= 00 için).

$\hat{y}_{i,1}\hat{y}_{i+1,1}\hat{y}_{i+2,1}$	000	001	010	011	100	101	110	111
$\hat{y}_{i,1}\hat{y}_{i+1,1}\hat{y}_{i+2,1}$	000	00**0**	0**0**0	**1**11	**0**00	1**1**1	11**1**	111

2.5. Hamming kodu. Bilgi transferinde bir bitlik hatayı bulup düzeltebilen Hamming kodunu bilgi gizleme tekniği olarak da kullanmak mümkündür [5]. Bu yöntemde gizlenecek bilgi p kadar gruplara ayrılır. Bu yöntemde imgedeki değişim oranı $\epsilon = \frac{p}{1-2^{-p}}$, bilgi gizleme kapasitesi ise $\alpha = \frac{p}{2^p-1}$'dir. Bu yöntemin algoritması aşağıdaki gibidir:

1. Gizlenecek bilgi p uzunluğunda gruplanır.
2. Gizlenecek grubu saklamak için örtü imgede $2^p - 1$ bir grup oluşturulur.
3. $p \times (2^p - 1)$ boyutunda bir Hamming matrisi oluşturulur.
4. Örtü imgeden ayrılan grubun transpozu ile Hamming matrisinin çarpımı hesaplanır.
5. Çarpım sonucundaki matris ile gizlenecek bit grubu aynı ise değişiklik yapılmaz.
6. Eğer aynı değiller ise çarpım sonucunda çıkan matris ile gizlenecek bit matrisi XOR'lanır.
7. Örtü imgeden ayrılan grupta 6. adımda çıkan sonucun onluk sistemde denk geldiği değerdeki bitin tümleyeni alınır.

Örnek olarak gizlenecek veri uzunluğu $p = 3$ olsun. Örtü imgeden $2^p - 1 = 7$ pikselin son bitleri alınır. Gizlenecek bit grubu $m = 110$ olsun. Örtü imgeden alınan grup $y_i = 6(0000110)$ olsun. 7 uzunluğundaki piksel grubu için oluşturulacak matris ve yukarıdaki algoritmaya göre yapılacak işlemler sonucu aşağıdaki gibidir.

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$H.y^T = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \neq \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Çıkan sonuca göre örtü imgedeki 7'li grubun dördüncü bitinin tümleyeninin alınması gerekir. Örtü imgedeki grubun yeni durumu $\hat{y} = 1000000$'dır. Gizlenen bilgiyi elde

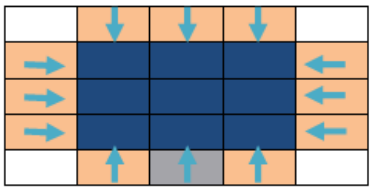
etmek için Hamming matrisi ile stego imgeden $2^p - 1$ uzunluğundaki grupların son bitleri ile çarpılır ve p uzunluğundaki gizlenen bilgiye ulaşılır.

1024 bitlik bilgiyi farklı uzunluğa (p bitlik uzunluklara) bölünmüş örtü imgede gizlemek için gerekli piksel sayısı ve değişime uğrayan piksel sayısı Tablo 6'da verilmiştir.

TABLO 6. Hamming matrisi ile farklı p uzunluklarında gruplarda 1024 bit gizlemek için gerekli piksel sayısı, n, değişen piksel sayısı l.

p	1	2	3	4	5	6	7	8	9	10
n	1024	1536	2389	3840	6348	10752	18578	32640	58140	104760
l	512	384	298.7	240	198.4	168	145.1	127.5	113.6	102.3
n/l	0.50	0.25	0.13	0.063	0.031	0.016	0.008	0.004	0.002	0.0009

2.6. İmge kareleri yöntemi. Bu yöntemde gizlenecek mesaj istenilen uzunlukta parçalara, p, ayrılır. Bu mesaj, m_i, parçasını saklamak için imge $n * n$ boyutunda karelere bölünür. Burada n, $p + 2$'ye eşittir. Şekil 2'de gösterildiği gibi $p \times p$'lik karenin içinde kalan $p * p$ lik koyu kısımda sağdan sola, soldan sağa, yukarıdan aşağıya, aşağıdan yukarıya tarama yapılarak gizlenecek mesaja en benzer grup bulunur, mesajla uymayan son bitler değiştirilir. Gizlenen mesajın işaretlenmesinde kullanılır. Gizlenen mesajın nereden başladığını belirtmek için çerçevedeki değerin son biti 1'den 0'a veya 0'dan 1'e değiştirilir [6].



ŞEKİL 2. $n * n$ boyutundaki imge karesi.

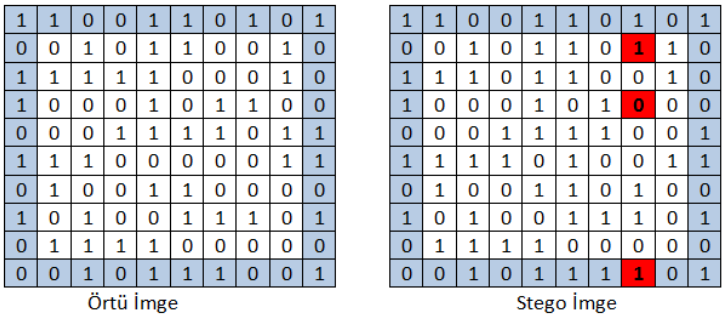
Gizlenen mesajın elde edilmesi için imge bilinen boyutlardaki karelere bölünür, her karede işaretlenen pikseller tespit edilir. Bu piksellerin belirttiği yöne göre mesajı içerin piksellerin son bitleri alınarak mesaja ulaşılır. Tablo 7'de gizlenecek mesaj bloğunun uzunluğuna göre gereken piksel sayıları belirtilmiştir. Tablodan da anlaşılacağı gibi mesaj blok uzunluğu arttıkça imgede saklanabilecek mesaj kapasitesi düşmektedir.

Örnek olarak, $m = 97 = 01100001$ olsun. Örtü imgede (bkz. Şekil 3) gizlenecek bilgi boyutlarını 8 bitlik parçalara ayıralım, bunun için örtü imge 9×9'luk bloklara ayrılır.

TABLO 7. İmge kareler yöntemiyle p bitlik mesajı gömmek için gerekli piksel sayısı (n).

p	2	3	4	5	6	7	8	9	10	11	12	13	14
n	16	25	36	49	64	81	100	121	144	169	196	225	256

Gizlenecek sekiz bitlik parça 9×9'luk parça içindeki 8×8'lik kısımda aşağıdan yukarıya, yakarıdan aşağıya, sağdan sola, soldan sağa aranarak en az değişiklikle gizlenebilecek satır veya sütun belirlenir. Daha sonra yönünü belirtecek işaret biti değiştirilerek gizleme işlemi tamamlanır. Bu örnek için $p = 8$, $n = 100$ olduğundan $\alpha = 8/100$, ve $\epsilon = 3/9$ dur.



ŞEKİL 3. İmge kareleri yöntemiyle bilgi gizleme örneği.

2.7. Piksel farkı yöntemi. Bu yöntemde örtü imgedeki ardışık gelen pikseller üst üste gelmeyen parçalara ayrılır. Her bloktaki iki pikselin değerinden fark değeri hesaplanır. Bütün olası fark değerleri farklı aralıklarla sınıflandırılırlar. Belirlenen bu aralıklar insan gözünün duyarlılığı göz önünde tutularak belirlenir. Daha sonra, bu fark değerinin yerine yeni bir değer, gizlenecek mesaj bitleri gömmek için yazılır [7].

$M \times N$ boyutunudaki F örtü imgesi art arta gelen piksellerle bloklara ayrılır:

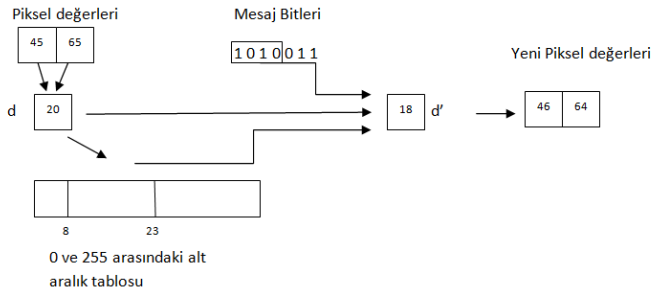
$$F = \left\{ F_i | i = 1, 2, \ldots, \frac{M \times N}{2} \right\}.$$

F_i, $P(i, x)$ ve $P(i, y)$ olmak üzere iki pikselden oluşur. İki piksel arasındaki fark hesaplanır: $d_i = |P(i, x) - P(i, y)|$. Eğer d, 0'a yakınsa düz (smooth) bir blok, -255 ya da 255 değerine yakınsa bir kenar (edge) bloğu olduğu düşünülür. d'nin mutlak değeri 0 ve 255 arasındaki değerleri, $R = \{R_i \,|\, i = 1, 2, \ldots, n\}$ alt aralıklara bölünür. Her alt aralığın bir alt ve üst değerleri vardır, $\{l_i\}$ ve $\{u_i\}$ öyle ki $l_1 = 0$, $u_n = 255$.

Bu alt aralıkların uzunlukları ikinin katı olacak şekilde şeçilir. Bu durumda, bu iki piksel arasına gizlenecek mesajın bit sayısı da $s = \log_2(u_i - l_i + 1)$ olacaktır. Fark değeri yerine mesaj bitlerine bağlı olarak yazılacak değer aynı alt aralıktaki başka bir fark değeri olacaktır. F_i'de gizlenebilcek mesajdan s bit okunup ondalık değerine çevrilir, $\{b\}$. Yeni fark değeri, $d_i^{'} = l_i + b$ dir.

$$\left(P'_{(i,x)}, P'_{(i,y)}\right) = \begin{cases} P_{(i,x)} + \left\lceil \frac{t}{2} \right\rceil, P_{(i,y)} - \left\lfloor \frac{t}{2} \right\rfloor, & \text{if } P_{(i,x)} \geq P_{(i,y)} \text{ and } d_i^{'} > d_i \\ P_{(i,x)} - \left\lfloor \frac{t}{2} \right\rfloor, P_{(i,y)} + \left\lceil \frac{t}{2} \right\rceil, & \text{if } P_{(i,x)} < P_{(i,y)} \text{ and } d_i^{'} > d_i \\ P_{(i,x)} - \left\lceil \frac{t}{2} \right\rceil, P_{(i,y)} + \left\lfloor \frac{t}{2} \right\rfloor, & \text{if } P_{(i,x)} \geq P_{(i,y)} \text{ and } d_i^{'} \leq d_i \\ P_{(i,x)} + \left\lceil \frac{t}{2} \right\rceil, P_{(i,y)} - \left\lfloor \frac{t}{2} \right\rfloor, & \text{if } P_{(i,x)} < P_{(i,y)} \text{ and } d_i^{'} \leq d_i \end{cases}$$

Yeni piksel değerlerini hesaplama kriterlerinde $t = |d_i^{'} - d_i|$'dir. Şekil 4 te bir örnek üzerinde bu yöntemin pikseller üzerinde meydana getirdiği değişiklikler gösterilmiştir. Bu örnekte, $d = 20$, $\{l_i = 8\}$ ve $\{u_i = 23\}$ alt aralığına düştüğünden ve $m_i = 1010$ olduğundan $d_i^{'} = 8 + 10$ ve $t = 2$'dir.



Şekil 4. Piksel farkı yöntemi için bir örnek.

2008 yılında Wang'ın [8] makalesinde modüler aritmetik yöntemi ile iki piksel arasındaki kalan fonksiyonunun sonucu kullanılarak piksel farkı yönteminin geliştirilmiş bir hali yayınlanmıştır. Tablo 8'de 256×256 boyutundaki bir imgeye, yüz baytlık veri gizlenerek bu iki yöntemin karşılaştırılması verilmiştir (her iki piksele üç bit bilgi gizlendiğinden kullanılan piksel sayısı 534'tür).

Tablo 8. Piksel farkı yöntemi ve Piksel modu yöntemlerinin karşılaştırılması.

Yöntem	Gizleme Verimliligi	Değişen Piksel Sayısı	PSNR	MSE
Piksel Farkı Yöntemi	1.2304	434	62.5182	0.0367
Piksel Modu Yöntemi[8]	1.335	400	67.3012	0.0122

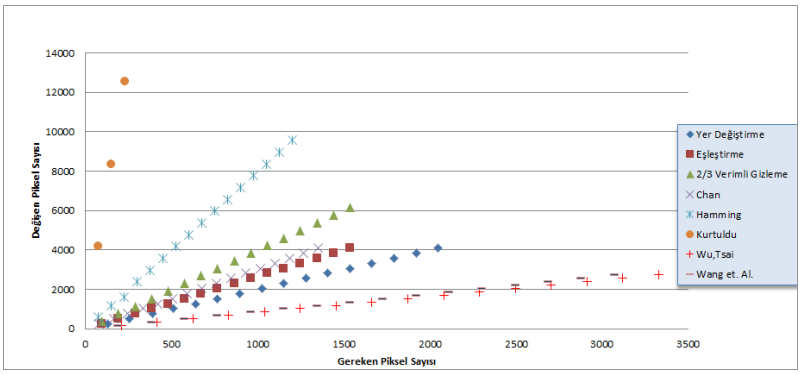
3. Sonuçlar

LSB yöntemlerinin karşılaştırılması amacıyla 256x256 boyutlarındaki gri tonlamalı bir imgenin içine 1850 bayt boyutundali bir veri yukarıdaki herbir yöntem kullanılarak gizlenmiştir ve PSNR ve MSE değerleri Tablo 9'da verilmiştir.

TABLO 9. $\alpha = 1/1$ kapasitesiyle bilgi gizleyen yöntemlerin imgede oluşturduğu değişiklikler.

Dosya boyutu (bayt)	Yöntem	Degistirilen Bit	Gizleme Verimliligi	PSNR	MSE
100	Mielikainen	286	2.7972	71.7319	0.0044
	Chan	255	3.1373	72.2302	0.0039
	Yer Değiştirme	368	2.1739	70.6371	0.0056
250	Mielikainen	730	2.7397	67.6624	0.0110
	Chan	658	3.0395	68.1133	0.0100
	Yer Değiştirme	941	2.1254	66.5597	0.0144
500	Mielikainen	1502	2.6631	64.5289	0.0229
	Chan	1321	3.0280	65.0866	0.0202
	Yer Değiştirme	1952	2.0492	63.3908	0.0298
1024	Mielikainen	3118	2.6273	61.3568	0.0476
	Chan	2729	3.0311	61.9356	0.0416
	Yer Değiştirme	4152	1.9730	60.1130	0.0643
1850	Mielikainen	5589	2.6480	58.8223	0.0853
	Chan	4926	3.0045	59.3707	0.0752
	Yer Değiştirme	7391	2.0024	57.6086	0.1128

Şekil 5'te ise makalede incelenen tüm yöntemler karşılaştırılmış ve imgede değişecek olan piksel miktarı ve bu bilgiyi gizlemek için gereken piksel sayısı verilmiştir.



ŞEKİL 5. Gizlenen bilgiye göre imgedeki değişen piksel miktarı ve gereken piksel sayısı grafiği.

Çeşitli steganaliz yöntemleri imgedeki toplam piksel sayısının en az 0.002'sinde değişiklik olduğu durumlarda veri gizlendiğini tesbit edebilmektedirler [17]. Buna

göre Tablo 10'da 1024 bitlik bilgiyi gizlemek için incelen yöntemlere göre örtü imgenin en az kaç pikselden oluşması gerektiği listelenmiştir.

TABLO 10. 1024 bitlik bilgi güvenilir bir şekilde gizlenmesi için gereken örtü imge piksel sayısı.

Yöntem	Yer Değiştirme	Eşleştirme	Chan*	2/3 Veri Gömme	Hamming**
Gerekli en az piksel	256000	192000	170500	288000	58140
Değişen piksel	512	384	341	384	113.5

*Chan'ın eşleştirme yöntemi için değişim oranı 3/1 olarak alınmıştır.
**Hamming yöntemi için Tablo 6 daki veriler kullanılmıştır.

İmge kareleri yönteminin başarısını belirlemek için yazarlar 256×256 boyutundaki gri tonlamalı imgelere 3584 bit uzunluğunda bir mesaj gizlenerek en uygun MSE değerini 14×14'lük karelerin verdiğini tespit ettiklerini ifade etmişlerdir [6], bu nedenle yapılan karşılaştırmalarda bu boyut kullanılmıştır. Örtü imgeyi 14×14'lük karelere bölüp, herbir kareye 12 bit uzunluğunda olmak üzere 3888 bit uzunluğunda mesaj gizlenmesi sonucu elde edilen MSE değeri 0.017593'tür. Verilen değere göre imgedeki $d \approx 1153$ pikselin değerinin ∓ 1 bit değiştiğini tespit edebiliriz. Bu sonuca göre incelenen diğer yöntemlerle karşılaştırılması Tablo 11'de verilmiştir.

TABLO 11. İmge Kareleri Yöntemi için gerekli piksel sayısı.

Yöntem	Değişen piksel sayısı	Gizleme için gerekli piksel sayısı
İmge Kareleri**	1153	18833
Mielikainen	1458	3888
Chan*	1285	3888
Hamming matrisi	911	14580

*Karşılaştırma için Chan' ın eşleştirme yöntemiyle 500 bayt gizleme verileri kullanılmıştır.
**Karşılaştırmada imge kareleri yöntemi ile Tablo 7 daki veriler incelenerek gizleme için gereken piksel sayısının en yakın olduğu Hamming matrisi olarak 4 seçilmiştir.

Bu sonuçlara göre Mielikainen ve Chan'ın yöntemleri ile bilgi gizlemek için yaklaşık beş kat daha az piksel gerekmesine rağmen imgedeki değişiklik imge kareleri yönteminden daha fazladır. Hamming matrisi ile imgede daha az piksel kullanarak daha az değişiklik sağlamak mümkündür. İmge Kareler yöntemi, Hamming matrisi kadar başarlı olmamakla birlikte en büyük zaafı ve bilginin gizliliğini tehdit eden unsur, alıcı tarafın imgenin orjinal haline veya bilginin gizlenme yönünü belirleyen işaretleme piksellerin bilgisine ihtiyaç duyulmasıdır.

Bu çalışmada imgelerde bilgi gizleme amaçlı kullanılan bazı temel LSB yöntemler incelenmiştir. Stego imgenin iletiminde kullanılacak yolun güvenilirliğine göre farklı yöntemler seçilebilir. Eğer gönderilecek veri miktarı büyük ve iletimin güvenli olduğu kabul ediliyorsa Chan'ın yöntemi bu çalışmada incelenen yöntemlerin en uygunudur. Fakat, iletim yolunun güvenliğinden şüphe varsa steganaliz yöntemlerinin imgedeki değişikliği farketmesi için gereken en fazla değişim oranının altında kalmak şartıyla örtü imgenin boyutlarına göre öncelikle Hamming kodlarını kullanarak geliştirilen yöntem ve sonrasında Chan'ın yöntemi olmak üzere tüm yöntemler kullanılabilir.

Kaynaklar

[1] N. F. Johnson and S. Katzenbeisser, A survey of steganographic techniques, in: *Information Hiding Techniques for Steganography and Digital Watermarking* (eds. S. Katzenbeisser and F. A. P. Petitcolas), Artech House Books (2000), 43–78.

[2] J. Mielikainen, LSB matching revisited, *IEEE Signal Processing Letters* **13** (2006), 285–287.

[3] C. S. Chan, On using LSB matching function for data hiding in pixels, *Fundamenta Informaticae* **96** (2009), 49–59.

[4] A. D. Ker, Information hiding and covert communication, `www.sti.uniurb.it/events/fosad08/slides/ker-slides-part2.pdf`. Online; accessed 26-February-2013.

[5] J. Fridrich and D. Soukal, Matrix embedding for large payloads, *Proceedings of the SPIE* **6072** (2006), 727–738.

[6] Ö. Kurtuldu ve N. Arıca, İmge kareleri kullanan yeni bir steganografi yöntemi, *Journal of Naval Science and Engineering* **5** (2009), 107–118.

[7] D. C. Wu and W. H. Tsai, A steganographic method for images by pixel-value differencing, *Pattern Recognition Letters* **24** (2003), 1613–1626.

[8] C. M. Wang, N. I. Wu, C. S. Tsaib and M. S. Hwang, A high quality steganographic method with pixel-value differencing and modulus function, *Journal of Systems and Software* **81** (2008), 150–158.

[9] B. Li, J. He, J. Huang and Y. Q. Shi, A survey on image steganography and steganalysis, *Journal of Information Hiding and Multimedia Signal Processing* **2** (2011), 142–172.

[10] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, Information hiding: A survey, *Proceedings of the IEEE* **87** (1999), 1062–78.

[11] H. K. Pan, Y. Y. Chen and Y. C. Tseng, A secure data hiding scheme for two-color images, *Proceedings of the Fifth IEEE Symposium on Computers and Communications (ISCC 2000)*, 750–755.

[12] N. F. Johnson, Z. Duric, and S. Jajodia, *Information Hiding: Steganography and Watermarking - Attacks and Countermeasures*, Kluwer Academic Publishers, 2000.

[13] D. Artz, Digital steganography: hiding data within data, *Internet Computing* **5** (2001), 75–80.

[14] A. Şahin, *Görüntü Steganografide Kullanılan Yeni Metodlar ve Bu Metodların Güvenilirlikleri*, Doktora Tezi, Trakya Üniversitesi, Edirne 2007.

[15] ShantalaSureh and Vishvanath, Edge-steganography for secure communication, *TENCON 2006, IEEE Region 10 Conference* (2006), 1–4.

[16] Kh. M. Singh, L. S. Singh, A. B. Singh and Kh. S. Devi, Hiding secret message in edges of the images, *International Conference on Information and Communication Technology (ICICT 2007)* (2007), 238–241.

[17] A. D. Ker, Quantitative evaluation of pairs and RS steganalysis, *Proceedings of the SPIE* **5306** (2004), 83–95.