EN DÜŞÜK BOZULMAYA SAHİP YENİ BİR LSB VERİ GİZLEME ŞEMASI

Teknoloijnin hızla gelişmesiyle, artan ihtiyaçlar, internetin erişilebilirliğinin kolaylığı, verilerin dijitalleştirilmesine ve internet üzerinden iletişimi normal hale getirmiştir. Bu sebeple bilgilerin güvenliği de önem kazanmıştır.

Veri gizleme / steganografi yöntemi, bahsettiğimiz düz metinlerin şifreleneceği, bir koruma tekniği kullanılır. Bu yöntemle bir görüntü kullanılarak, gizli bir mesajı gizleme durumu söz konusudur. İletişim bu şekilde yürütülür ve tamamen tespit edilmesi çok mümkün değildir.

En eski endişe mahkum Alice ve Bob arasında, gardiyan Wendy'e yakalanmadan kaçış planı yapabilmeleri için, aralarındaki şifreli mesaj ile başlamıştır.

Günümüze kadar, gizli mesajları iletebilmek için, resim, video, ses gibi örüntü nesneler kullanılmıştır.

Veri gizleme için birçok yöntem ve algoritma bulunmaktadır. Temelde bir veri gizleme şeması iki performans tekniği ile değerlendirilir.
Birincisi gizleme esnasında oluşan algısal bozulma, diğeri ise gömülebilir kapasitedir.

PSNR, yani *tepe/sinyal/gürültü oranı* işlenmiş olan görüntü ve videoların, görsel bozulmasını değerlendirmek için kullanılmaktadır.

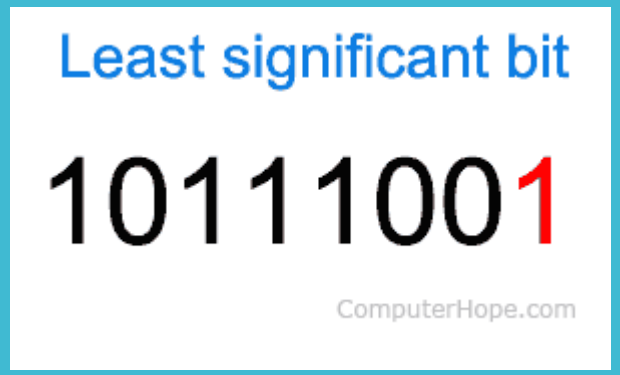
Gömülebilir kapasite işlenirken ise, herhangi bir gözle görülebilir bozulma olmadan, kapak görüntüsüne maksimum sığacak gizli mesaj miktarını gösterir.

En eski ve en ünlü veri gizleme tekniği **LSB** (*Least Signification Bit*)'ler yöntemidir. Yani en az anlamlı, en az öneme sahip bit tekniği olarak geçmektedir.

Uygulaması çok basit bir yöntemdir. Mesaj gizlenirken, insan duyarlılığına duyarlı olmayan bir piksel için, en önemsiz bitler varyasyonu kullanılır. Teknik, diğer multimedya nesneleri içinde rahatlıkla kullanılmaktadır.

Bu algoritma ile ilgili ilk çalışmaları Wang ve arkadaşları, karşıt bir görüş olarak değil de, geliştirmeye yönelik çalışmaları da Chang ve diğerleri yapmıştır. LSB yöntemi için, optimal düzeyde maliyet hesaplamaları ve görsel kalite çalışmaları için, iyileştirmelerde bulunmuşlardır.

Yapılan çalışmalar tam olarak, beklentiyi sağlamadığı için, Mielikainan yeni bir yöntem ortaya atmıştır. Bir iki fonksiyona dayalı gizli verinin iki bitini gizlemek için, bir piksel çifti kullanılır. Böylece, az önce bahsettiğimiz kişilerde yaşanabilecek görsel bozulma, daha aza indirilmiş olur.



İnceleyeceğimiz bu makalede, basit düzeydeki bir LSB yöntemi, optimal düzeyde LSB yöntemi, eşleştirme yeniden ziyaret yöntemi, matematiksel alanda teorik analizler, bazı deneysel sonuçlar ve LSB yerleştirme şemalarına yer verilecektir :

Basit LSB Yöntemi

Bu şemanın ilk uygulamaları, gri tonlamalı örüntüler içindir. Daha kolay ve erişilebilir olduğu için, basit bir gömme sürecinden geçmektedir. İsminin bu şekilde anılma sebebi de bu olabilir.

Gri seviyeli piksel değeri, LSB düzlemini doğrudan değiştirme yoluyla etkiler, mesajlar da bu şekilde kolaylıkla taşınabilir. İnsan görsel duyarlılığı açısından görünmez olduğu için, verinin taşınmasını mümkün kılar.

Gri tonlamalı bir pikselin ondalık değerinin G(10), sekiz bitlik ikili dosyaya dönüştürülmesi:

G(10)= d8d7- d6d5d4d3d2d1 ile yazılabilir.

Bir örnek ile gösterecek olursak, **G(10)= 28(10)** ikili değeri 00011100'dır. *0010* gibi dörtlük bir mesajı gömmek istiyorsak, en doğru ve en az anlamlı *1100* bitlerini doğrudan değiştirerek, pikselin orijinal ikili gösterimini *00010010* haline getiririz. Bu haldeyken ondalık değer **18(10)** olur.

Örneğe benzer şekilde, n bit gizli mesaj gizleme durumunda, bir gizli mesajında n bitini kaplayabilmek için, her pikselin en az anlamlı n bitini kullanmak durumunda kalırız.

Ayrıca, bir pikselin LSB düzlemine kaç bit gizli mesaj gömülü olduğu bilgisine sahipsek, bir pikselin en az önemli bilgilerini okuyarak, gömülü gizli mesajı da çözümleyebiliriz.

Optimal LSB Yöntemi

Basit LSB şemalarındaki bozulma miktarını azaltabilmek için öne sürülen çalışmalardan bir tanesidir. Gizli mesajları bir görüntünün bir pikseline gömdüğümüz durumlarda, basit LSB şeması yalnızca bir stego piksel değeri üretmektedir. Optimal LSB şeması ise, biri en az bozulmaya sahip olan 3 adet stego piksel değeri üretir.

Doğal olarak gizli mesajlar taşınırken, bu yol tercih edilir. Çünkü bir stego görüntüsünün kalitesi önemli ölçüde etkilemektedir.

Orijinal piksel değeri **G(10)= 28(10)** olan, 4 bit gizli mesajı **0010(2)** piksele gömmek istediğimizde,
R = G(10)mod 2n hesaplama yöntemiyle, kalanı buluruz. Bu durumda ***R=12*** ve gizli mesajımız 2'ye eşittir. Bu noktaya gelene kadar gerekli ayarlamalar yapılır. Mesajlar, orijinal piksel değerinin en sağdaki dört bitine doğru gömülür: **28(10)= 00011100(2)** halini alır.

Bu işlemlerden sonra, stego piksel değerini 18 elde ederiz. **18(10) = 00010010(2)** kalan değerini, ondalık değerine eşit bir mod alarak, stego piksel değeri için +2 ve -2 değerleri ile hesaplarız. Yani **18(10)+2 n ve 18(10)-2 n** ile sırasıyla 34 olan diğer iki stego piksel değerini elde etmek için aynı işlemler yapılır. [***(34 mod 24= 2) ve 2(10)(2 mod 24= 2)***]

Buna göre sonuçta 18 olan üç stego piksel adayımız olur.Üç aday arasından (***34-28=6***) en az bozulmaya sahip olan seçilir.

Ancak, basit LSB şeması kullanılırsa ***18(10)*** tek seçenek, mesafesi de (yani piksel bozulması) 10 olur. Optimal LSB şemasının, basit LSB şeması bozulmasını büyük ölçüde azalttığı bu işlemler ile gözlenebilir.

Tablo 1. Basit LSB yöntemi ile optimal LSB şeması arasındaki görsel bozulmanın karşılaştırılması.

Kapak resmi (512 × 512)	n-LSB'ler (kapasite: 512 × 512 ×n)	PSNR (dB) Basit LSB'ler şema	PSNR (dB) Optimal LSB'ler şema
Lena	n =1	51.12	51.12
	n =2	44.02	46.37
	n =3	37.86	40.72
	n =4	31.28	34.84
babun	n =1	51.14	51.14
	n =2	44.02	46.37
	n =3	37.86	40.72
	n =4	31.33	34.78

Yan taraftaki tabloyu incelediğimizde, optimal LSB yönteminin, her pikselin gömme kapasitesinde n=1 bit hariç, basit LSB yönteminde bozulma yaşanabileceği ve bunu da 2-3 db PSNR ile azaltabileceğini görmekteyiz.

Eşleştirilmiş Revize Yöntemi ile LSB

Basit LSB şeması veya optimal LSB şeması, bir gömme birimi gömme işleminde yalnızca bir pikselden oluşur. Yani, bir bit gizli mesajın bir biti sırayla bir piksele gizlenir ve bu nedenle, piksel başına sadece bir bit gizli mesaj gizleyebiliriz.

Eşleştirilmiş Revize Yöntemi ile LSB , iki pikseli ikili bir işleve dayalı bir gömme birimi olarak gruplandırır. Bu işlemin konsepti, gri seviyeli bir görüntüyü örtüşmeyen piksel çiftlerine bölmek ve ardından her piksel çiftine iki bit gömmek için ikili bir işlev kullanmaktır.

Bu şemanın genel mantığı , yalnızca +1 veya -1 pikselini değiştirerek iki biti bir piksel çiftinde gizleyebilir. Geleneksel veri gömme işleminde, veri gizledikten sonra en fazla bir piksele bir bit ekleyebilir veya bir piksel çiftinin bir pikselinden bir bit çıkarabiliriz. Bu nedenle birden fazla pikseli değiştirmek neredeyse imkansızdır. Piksel değişikliği olasılığı azaldığından, piksel başına beklenen değişiklik sayısı da 0,5'ten 0,375'e düşürülmüştür. Bu sonuç, veriler gizlendikten sonra görsel bozulmanın korunabileceğini göstermektedir. Eşleştirilmiş Revize Yöntemi ile LSB şemasının gömme algoritmasını inceleyelim

- mi ve mi+1(sırasıyla gizli mesajın iki biti) tanımlanır
- yi ve yi+1 sırasıyla kapak piksel çiftleridir.
- bit mi LSB ile gizlenir
- stego piksel yi düzlemi ve mi+1 biti gizlenir
- yi ve yi+1 piksel çiftinin fonksiyonuna dönüştürülür.
- ikili fonksiyonun denklemi aşağıda verilmiştir.

$$\begin{cases} \mathrm{LSB}(y_i) = m_i \\ m_{i+1} = f(y_i, y_{i+1}) = \mathrm{LSB}\left(\left\lfloor \frac{y_i}{2} \right\rfloor + y_{i+1}\right) \end{cases}$$

Eşleştirilmiş Revize Yöntemi ile LSB için bir örnek yapalım :

- Orijinal piksel çiftinin ondalık değerlerinin *100(10) ve 51(10)* ve ikili tabandaki değerleri sırasıyla 01100100(2) ve 00110011(2) olarak belirleyelim.
- Ardından mi = 0 mi+1 = 0 değerlerini varsayım olarak belirleyelim.
- mi = 0, mi+1 = 0 değerini 100(10) ve 51(10) piksel değerinin içine gizleyelim.
- 51değerini, 50 yada 52'ye nasıl değiştirirsek değiştirelim, 100'ün değişmediğini görüyoruz.
- mi = 0 ve mi+1 = 1 değerini verdiğimiz durumda iki piksel değerinde değişiklik yapmamıza gerek yoktur.
- Eğer mi = 1 , mi+1 = 0 olursa 100 değerini 99'a çevirmemiz gerekiyor ve 51 sayısını değiştirmemize gerek yoktur.
- Son olarak mi = 1 ve mi+1 = 1 değerini aldığımızda 100 sayısı 101 olarak değişir ve 51 sayısında değişim olmaz.

Original pixel pair	Secret message (m_i, m_{i+1})	Pixel variation
$f(100,51)$	(0,0)	$f(100,51-1)$ or $f(100,51+1)$
$f(100,51)$	(0,1)	No modify
$f(100,51)$	(1,0)	$f(100-1,51)$
$f(100,51)$	(1,1)	$f(100+1,51)$

Cover image (512 × 512)	Capacity (bits)	PSNR (dB)		
		Simple LSB	Optimal LSB	LSB matching
Lena	262144	51.12	51.12	52.41
Baboon	262144	51.14	51.14	52.41

Yandaki tabloda Eşleştirilmiş Revize Yöntemi ile LSB'ye göre bir gömme birimi için tüm durumları göstermektedir.

- İki bit gizli mesaj taşıyan bir çiftin iki pikselini aynı anda değiştirme durumu yoktur.
- Değiştirilmesi gereken bir piksel çiftinin olasılığı 3/4 ve korunan orijinal piksel çiftinin olasılığı 1/4'tür.
- Yeniden ziyaret edilen LSB eşleştirmesi tarafından üretilen bir piksel çiftinin piksel başına modifikasyonunun (ENMPP) beklenen sayısı (1/4) × 0 + (3/4) × 1 = (3/4). Ortalama olarak (3/4)/2 = 0.375 ile her pikselin ENMPP'sini daha da hesaplayabiliriz.
- Sonuç olarak Eşleştirilmiş Revize Yöntemi ile LSB şemasının, ENMPP açısından basit ve en uygun LSB şemasının performansını artırır.
- Yandaki tabloda da görüldüğü gibi Eşleştirilmiş Revize Yöntemi ile LSB'nin PSNR'si optimal ve basit LSB yönteminden daha iyi bir durumdadır.

Önerilen Şema

Makalenin bu kısmında üç pikseli bir piksel çifti yerine bir gömme birimi olarak gruplayarak Eşleştirilmiş Revize Yöntemi ile LSB'nin performansını iyileştirmek için yeni bir algoritmayı önermektedir.

- Gizli bir mesajın durumunu kaydederken ikinci en az anlamlı biti kontrol etmek için +1 veya-1 operatörünü kullanırız.
- Gömme ünitesinden üç orijinal bit elde etmek için XOR işlemi ile altı en az anlamlı bit birleştirilir.En fazla bir pikselin +1 veya -1 ile değiştirilebilmesini sağlamak için yeni bir gömme algoritması sunulmaktadır.
- Yani, gizlenecek verileri gizlemeden önceki ve sonraki piksel değerinin maksimum değişimi +1 veya -1 piksel fark değeri ile sınırlıdır.
- Önerilen şema ile aynı zamanda üç pikselden oluşan bir gömme birimini göstermek için bit modifikasyonunun kullanılmasının kalite bozulması açısından en iyi optimal kombinasyon olduğu da ortaya çıkarılmıştır.

Şemanın genel olarak mantığını açıklamamız gerekirse

- Üç pikseli pi, pi+1, pi+2 tanımlarız ve bu pikselleri bir gömme biriminde gruplarız.
- Daha sona üç bit gizli veri seçeriz ve bunları pi, pi+1, pi+2 içine gizleriz.
- Gömme algoritmalarını aşağıda kısaca açıkladık.

ADIM 1

pi, pi+1, pi+2 ondalık değerlerini binary değere dönüştürürüz

- pi(10) = a8 a7 a6 a5 a4 a3 a2a1(2),
- pi+1(10) = b8 b7 b6 b5 b4 b3 b2 b1(2)
- pi+2(10) = c8c7 c6 c5 c4 c3 c2 c1(2),

- a1, pi‘nin ilk en doğru biti olsun
- a2 pi‘nin en doğru ikinci biti olsun
- Aynı şekilde tanımladığımız pi+1 ve pi+2'nin ilk en sağ biti olarak b1 ve c1
- pi+1 ve pi+2'nin sağ biti b2 ve c2'yi en çok ikinci olarak tanımlayın
- Birinci ve ikinci en sağ bitlerin varyasyonu üç pikseli +1 veya -1 ile değiştirerek kontrol ederiz.
- Örneğin, pi = 10(10) = a8 a7 a6 a5 a4 a3 a2 a1(2) = 00001010(2), a1 = 1 ile pi +1 veya pi -1.
- Bu durumda eğer a2 0 ise, pi'yi sadece -1 ile değiştirebiliriz.
- Şema için yeni plan, üç pikselden oluşan bir gruba XOR işlemi uygulanarak sağlanır.

- Ayrıca, A, B ve C bitlerini bir gömme biriminin orijinal özellik değerleri olarak tanımladığımız için, önerilen gömme algoritmasını uyguladıktan sonra veriler gizli anahtarın üç bitine dönüştürülecektir.
- Spesifik olarak, A, B ve C bitlerinin önemi, gizli bir mesajın durumunu üç ikili bit şeklinde gizlemek için kullanılacaktır.

ADIM 2

Algoritmanın 2. adımında, A, B ve C bitlerini oluşturma kuralı ayrıntılı olarak açıklanmaktadır.

- Denkleme göre, en sağdaki altı anlamlı biti XOR yaparak A, B ve C bitlerini hesaplarız.

$$\begin{cases} A = a_1 \oplus a_2 \oplus b_1 \\ B = b_1 \oplus b_2 \oplus c_1 \\ C = c_1 \oplus c_2 \oplus a_1 \end{cases}$$

- Denklemde gösterildiği gibi, pi'nin en az anlamlı bitini a1 değiştirerek A ve C bitlerini aynı anda kontrol edebiliriz.
- Aynı şekilde, bit A, pi'nin ikinci en az anlamlı biti a2 değiştirilerek kontrol edilebilir.
- Spesifik olarak, piksel değeri pi tek bir sayı olduğunda, onu yalnızca a1 bitini pi -1 ile değiştirerek kontrol etmemiz gerekir.
- Piksel değeri pi bir çift sayı olduğunda, onu sadece a1 bitini pi +1 ile değiştirerek kontrol etmemiz gerekir.
- Benzer şekilde, eğer pi piksel değeri çift bir sayı ise, bit a2'yi sadece pi +1 ile kontrol etmemiz gerekir.
- Aksine, pi piksel değeri tek bir sayıysa, sadece bit a2'yi pi -1 ile kontrol etmemiz gerekir.

- Daha sonra, pi+1'in en az anlamlı b1 bitini değiştirerek A ve B bitlerinin durumunu eşzamanlı olarak değiştirebiliriz.
- Son olarak, B ve C bitleri, pi+2'nin en az anlamlı bit c1'i değiştirilerek aynı anda değiştirebiliriz.
- Bit C, pi+2'nin ikinci en az anlamlı bit c2'si değiştirilerek değiştirilebilir.

ADIM 3

- Çıkarılan orijinal özellik değeri bitleri A, B, C'yi m0, m1, m2 gizli mesajının üç biti ile aynı olup olmadıklarını görmek için karşılaştırırız.
- (A,B,C)2 = (m0, m1, m2)2 koşulu sağlanırsa, veri gömme işleminde pi, pi+1, pi+2 piksellerinin değiştirilmesi gereksizdir.
- Aksi takdirde, (A,B,C)2 = (m0, m1, m2)2 koşulu sağlanana kadar pi, pi+1 veya pi+2 piksellerini değiştirmeliyiz.

- pi , pi+1, pi+2 piksellerinin varyasyonunun ±1 ile sınırlandırılmasını sağlamak için modifikasyon algoritmasını öneriyoruz.
- Sonraki sayfada , algoritma ayrıntılı olarak verilmiştir.

If $(A=m_0)$ && $(B=m_1)$ && $(C=m_2)$
    No Modify
Elself $(A \neq m_0)$ && $(B=m_1)$ && $(C=m_2)$
    If p_{i+1} mod 2=0 Then $p_{i+1}=p_{i+1}-1$
    Else $p_{i+1}=p_{i+1}+1$
End
Elself $(A=m_0)$ && $(B\neq m_1)$ && $(C=m_2)$
    If p_{i+2} mod 2 =0 Then $p_{i+2}= p_{i+2}-1$
    Else $p_{i+2}= p_{i+2}+1$
    End
Elself $(A=m_0)$ && $(B=m_1)$ && $(C\neq m_2)$
    If p_i mod 2 ≡ 0 Then $p_i \equiv p_i - 1$
    Else $p_i = p_i + 1$
    End
Elself $(A\neq m_0)$ && $(B\neq m_1)$ && $(C=m_2)$
    If p_{i+1} mod 2=0 Then $p_{i+1} \equiv p_{i+1} + 1$
    Else $p_{i+1} \equiv p_{i+1} - 1$
    End
Elself $(A =m_0)$ && $(B \neq m_1)$ && $(C\neq m_2)$
    If p_{i+2} mod 2 =0 Then $p_{i+2} = p_{i+2} + 1$
    Else $p_{i+2} = p_{i+2} - 1$
    End
Else If $(A\neq m_0)$ && $(B=m_1)$ && $(C\neq m_2)$
    If p_i mod 2 =0 Then $p_i \equiv p_i + 1$
    Else $p_i \equiv p_i - 1$
    End
Elself $(A\neq m_0)$ && $(B\neq m_1)$ && $(C\neq m_2)$
    If p_{i+2} mod 2 =0 Then $p_{i+2} = p_{i+2} - 1$
    Else $p_{i+2} = p_{i+2} + 1$
    End
    If p_i mod 2 =0 Then $p_i = p_i + 1$
    Else $p_i = p_i - 1$
    End
End

$$\begin{cases} A = a_1 \oplus a_2 \oplus b_1 \\ B = b_1 \oplus b_2 \oplus c_1 \\ C = c_1 \oplus c_2 \oplus a_1 \end{cases}$$

- Algoritmaya göre m0, m1, m2 gizli verisinin üç bitinin sırasıyla pi, pi+1, pi+2 üç pikseline gömülmesi sağlanır.
- Stego görüntüsü alındıktan sonra, gizli mesaj, kapak görüntüsü bilgisi bilgisi olmadan denklem ile çıkarılabilir.

Örneğin, soldaki tabloda pi= 51, pi+1 = 99 ve pi+2 = 33 olduğunda, elimizde a1 = 1, a2 = 1, b1 = 1, b2 = 1, c1 = 1 ve c2 = 0 olur . Orijinal özellik biti A'yı A = a1 ⊕ a2 ⊕ b1 aracılığıyla denkleme dayanarak hesaplanır ve bit A = 1 elde ederiz. Sonra, biti de türetebiliriz. B = 1 yoluyla B = b1 ⊕ b2 ⊕ c1 Son olarak, orijinal özellik biti C = 0 denklem ile hesaplanabilir.

Gizli verinin durumu m0m1m2 = 110(2) ise, m0 m1 m2 ABC = 110(2) sağlandığından pi , pi+1, pi+2 pikselleri arasında herhangi bir pikseli değiştirmek gereksizdir.

	Three cover pixels $p_i = 51$, $p_{i+1} = 99$, $p_{i+2} = 33$ The original feature $A = 1$, $B = 1$, $C = 0$		
Status of secret data	Stego pixel p_i	Stego pixel p_{i+1}	Stego pixel p_{i+2}
$m_0m_1m_2 = 000_{(2)}$	51	99−1 = 98	33
$m_0m_1m_2 = 001_{(2)}$	51−1 = 50	99	33+1 = 34
$m_0m_1m_2 = 010_{(2)}$	51	99+1 = 100	33
$m_0m_1m_2 = 011_{(2)}$	51−1 = 50	99	33
$m_0m_1m_2 = 100_{(2)}$	51	99	33+1 = 34
$m_0m_1m_2 = 101_{(2)}$	51	99	33−1 = 32
$m_0m_1m_2 = 110_{(2)}$	51	99	33
$m_0m_1m_2 = 111_{(2)}$	51+1 = 52	99	33

- Aksi takdirde, m0m1m2 = ABC sağlanana kadar bir veya iki pikseli +1 veya -1 ile değiştirmeliyiz. Tabloda , A, B ve C bitlerinin m0m1m2 ile aynı olması için pikselleri değiştirmenin yedi örneğini göstermedir

	Three cover pixels $p_i = 51$, $p_{i+1} = 99$, $p_{i+2} = 33$ The original feature $A = 1$, $B = 1$, $C = 0$		
Status of secret data	Stego pixel p_i	Stego pixel p_{i+1}	Stego pixel p_{i+2}
$m_0m_1m_2 = 000_{(2)}$	51	99−1 = 98	33
$m_0m_1m_2 = 001_{(2)}$	51−1 = 50	99	33+1 = 34
$m_0m_1m_2 = 010_{(2)}$	51	99+1 = 100	33
$m_0m_1m_2 = 011_{(2)}$	51−1 = 50	99	33
$m_0m_1m_2 = 100_{(2)}$	51	99	33+1 = 34
$m_0m_1m_2 = 101_{(2)}$	51	99	33−1 = 32
$m_0m_1m_2 = 110_{(2)}$	51	99	33
$m_0m_1m_2 = 111_{(2)}$	51+1 = 52	99	33

(Üç piksel pi , pi+1, pi+2'den oluşan bir gruba en fazla +1 veya −1 değiştirilerek üç bit m0, m1, m2'nin nasıl gömüleceğini gösteren bir örnek)

ANALİZ

Örnek tablolarda da incelediğimiz gibi örtü piksellerinin değişiklik varyasyonu, +1 veya -1 ile sınırlandırılabilmekte, iki pikseli aynı anda değiştirirken olasılığın sadece 1/8 olduğunu göstermekte.

Gizli bir veriyi gizlemek içinse, bir pikselin +1 veya -1 değiştirme olasılığıda en fazla 6/8 olmakta.

Bu sonuçlar dikkate alındığında, önerilen şema için, veri gizlendikten sonra piksel bozulmasının iyi bir şekilde korunmuş olduğunu gözlemekteyiz.

Önerilen şema, bir piksel +1 veya -1 işlemini değiştirerek durumu kaydedebilir ve ayrıca iki piksel aynı anda +1 veya -1 ile değiştirilirken durum tekrar incelenebilir. Aynı şekilde, aynı anda üç piksel değiştirildiğinde de bu durumu elde edebiliriz.

Bir kayıp olmaksızın, birden fazla piksel aynı anda değiştirildiğinde çok sayıda durum elde edilebilir. Bununla birlikte, gizli mesaj, gizleme işleminden sonra daha fazla görsel bozulmaya neden olabilir.

Bu nedenle, bir gömme birimi üçten fazla pikselden oluştuğunda yalnızca bir pikseli veya iki pikseli aynı anda değiştirerek durumu analiz edebiliriz.

Ayrıca, önerilen şema üç pikseli bir gömme durumu için grupladığında en uygun çözümü ve veri gizlendikten sonra görüntü bozulmasını nasıl önleyebildiğini de göstermektedir.

Deneyler ve Karşılaştırmalar

Önerilen LSB veri gizleme algoritmamızın performansını değerlendirmek için bu deneyde, iki yüz kapak görüntüsü alınır.

Burada, 512 × 512 piksel boyutunda üç tipik görüntü ile deneysel sonuçlar incelenecek :

Orta düzeyde görüntü karmaşıklığına sahip '***Lena***', yüksek doku özelliğine sahip '***Baboon***' ve göreceğimiz gibi daha yumuşak özelliklere sahip '*Jets*'.

(a) Lena (Cover image)

(b) Lena (Stego image)
PSNR=52.916 dB

(c) Baboon (Cover image)

(d) Baboon (Stego image)
PSNR=52.917 dB

(e) Jets (Cover image)

(f) Jets (Stego image)
PSNR=52.925 dB

Yanda da görüldüğü gibi, kalite bozulmasının incelenmesi için, PSNR metriği kullanılmıştır. Eğer PSNR değeri yüksek olursa, insan görsel duyarlılığı için algılanamaz olduğu ifadesi söylenebilir.

Gizli veriler gizlendikten sonra daha fazla bozulma olduğunda, bunun tersi olacağı ifadesine de ulaşılabilir.

b,d,f şekillerinde gördüğümüz üzere, **(512x512/3)x3=262143** bit gömme kapasitesi olduğunu hesaplayabiliriz. Verilen PSNR değerlerinde, önerilen şema ve sözde rastgele sonuçlar 1000 kez çalıştırıldığında ortalama değerler gözlenmektedir.

Bu nedenle, PSNR değerleri, doğru olmakla birlikte, önerilen şemanın beklenenden yüksek performans gösterdiğini söylemektedir.

Cover image	PSNR		
	LSB-based scheme [1,15,17,19,20]	LSB matching revisited scheme [18]	The proposed scheme
Lena (512 × 512)	51.156 dB	52.404 dB	52.916 dB
Jets (512 × 512)	51.143 dB	52.4 dB	52.925 dB
Baboon (512 × 512)	51.161 dB	52.405 dB	52.917 dB
Elaine (512 × 512)	51.17 dB	52.407 dB	52.914 dB
Man (512 × 512)	51.138 dB	52.39 dB	52.911 dB
Average	51.154 dB	52.401 dB	52.917 dB

Önerilen şema ile daha önceki algoritmaların sonuçları karşılaştırılmıştır. Yeni şema performansı, tüm LSB şemaları için en az bozulmaya sahip olduğunu göstermektedir.

Hesaplama karmaşıklığı toplam maliyeti hesaplanırken, makalemizin, önceki kısımlarında yapılan işlem ve yazılan algoritmalarda gözüktüğü üzere, sekiz farklı adıma ihtiyaç olduğu sonucuna varılır.

Bir pikselin ortalama maliyeti, $O(n) = 2.66$ olarak hesaplandığında, şema ile hesaplama karmaşıklığının toplam maliyeti, mevcut şemalara da eşit olduğu gözlenir.

Böylece bu şema, tüm gizli veri gömme ve çıkarma işlemlerinde doğrusal olarak kullanılabilir ifadesi çıkarılabilir.

Sonuçlar

Bu yazı ile Eşleştirilmiş Revize Yöntemi ile üretilen görüntü kalitesinin bozulmasını en aza indirecek, yeni bir şema öne sürülmüştür.

Üç bit gizli mesaj verisini iletmek için üç pikselli bir gömme sisteminde gruplandırılır ve LSB Revize şemasında kullanılan bir piksel çiftinin yerini alır. Her bir gömme birimi için, orijinal halini tanıtmak amacıyla altı biti birleştirmek için, bir pikselin en doğru iki biti çıkarılır. XOR işlemi kullanılarak, üç bitlik ikili biçimde gizli veriler oluşturulur.

Bu şema ile piksel başına beklenen modifikasyon sayısının iyileştirilebileceği gözlenmektedir. Üç pikselli bir gömme biriminin, gömme etkisinden kaynaklanan toplam görsel bozulmayı en aza indirebilen en optimal kombinasyon olduğu da kanıtlanmıştır.