Pamukkale Univ Muh Bilim Derg, 27(2), 96-113, 2021

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

24-bit renkli imge içine 24-bit renkli imge gizleyen yüksek kapasiteli düşük bozulumlu tersinir kayıplı yeni bir veri gizleme yöntemi (YKKG)

A new data hiding method with high capacity, low distortion, and reversible loss that hides 24-bit color image into 24-bit color image (YKKG)

Ali DURDU[1*]

[1]Yönetim Bilişim Sistemleri, Siyasal Bilgiler Fakültesi, Ankara Sosyal Bilimler Üniversitesi, Ankara, Türkiye.
ali.durdu@asbu.edu.tr

Geliş Tarihi/Received: 13.12.2019 Düzeltme Tarihi/Revision: 07.04.2020 doi: 10.5505/pajes.2020.50215
Kabul Tarihi/Accepted: 29.04.2020 Araştırma Makalesi/Research Article

Öz

Bu çalışmada, 24-bit renkli imge içine 24-bit renkli imgeyi kayıplı gizleyen yüksek kapasiteli, düşük bozulumlu ve tersinir yeni bir veri gizleme yöntemi (YKKG) önerilmiştir. Önerilen yöntem gizlenecek 24-bitlik imgeyi, 4-bitlik parçalara böler ve her parçayı 2-bitlik gizleme koduna indirgeyerek gizler. Bu şekilde 4-bitlik parça 2-bite indirgendiği için yöntem kayıplı gizleme yapmaktadır. 2-bitlik gizleme kodları 2-baytlık bloklara gizlenir. Geri çıkarma işleminde ise yöntem, 24-bit imge gizlenmiş 24-bit stego imgeden, sırasıyla 2-baytlık bloklardaki 2-bitlik gizleme kodlarını kullanarak 4-bitlik parçalar elde eder ve parçalar birleştirilerek 24-bitlik gizli imge tersinir olarak geri elde edilir. Yöntem gizlenecek verinin boyutunu yarı yarıya düşürdüğü için geleneksel LSB yöntemlerine göre iki kat kapasite sunmaktadır. Aynı oranda veri gizlendiğinde ise yöntem, örtü imgede geleneksel LSB yöntemlerine göre daha düşük bozulma oluşturmaktadır. Önerilen yöntemin imge kalitesini ölçmek için literatürde sıklıkla kullanılan tepe sinyal gürültü oranı (PSNR) ve yapısal benzerlik kalite ölçütü (SSIM) kullanılmıştır. Ayrıca önerilen yöntemin görsel ataklara karşı dayanıklılığını ölçebilmek için salt & pepper, gaussian, speckle ve poisson saldırı atakları kullanılmıştır. Test sonuçları önerilen yöntemin, geleneksel LSB yöntemine ve literatürdeki çalışmalara göre kapasite olarak iki kat daha verimli, algılanamazlık olarak ise daha yüksek PSNR ve SSIM değerleri elde ettiğini göstermiştir.

Anahtar kelimeler: Veri gizleme, Steganografi, Yüksek kapasiteli, Kayıplı, Tersinir, Düşük bozulumlu.

Abstract

In this study, a new high-capacity, low-distortion and reversible data hiding method (YKKG) that hides a 24-bit color image in a 24-bit color image is proposed. The proposed method divides the 24-bit image to be hidden into 4-bit pieces and hides each piece by reducing it to the 2-bit hide code. In this way, since the 4-bit piece is reduced to 2-bit, the method performs lossy concealment. 2-bit hiding codes are hidden in 2-byte blocks. In the undo process, the method obtains 4-bit pieces from the 24-bit image hidden from the 24-bit stego image, using 2-bit hiding codes in 2-byte blocks, respectively, and the parts are merged back into reversible 24-bit image. Since the method reduces the size of the data to be halved, it offers twice the capacity compared to traditional LSB methods. When the same amount of data is hidden, the method creates lower distortion in the cover image than traditional LSB methods. Peak signal to noise ratio (PSNR) and structural similarity quality criterion (SSIM), which are frequently used in the literature, were used to measure the image quality of the proposed method. In addition, salt & pepper, gaussian, speckle and poisson attack attacks were used to measure the resistance of the proposed method to visual attacks. The test results showed that the proposed method achieved twice as efficient capacity and higher PSNR and SSIM values than the traditional LSB method and the studies in the literature.

Keywords: Data hiding, Steganography, High capacity, Lossy, Reversible, Low distortion.

1 Giriş

Güvenli iletişim, kişisel güvenlik için önemli olduğu gibi devlet kurumlarına ait bilgilerin de güvenliği içinde önemlidir. Güvenli iletişim yöntemlerinden steganografi alanında birçok çalışma bulunmaktadır. Veri gizleme işleminde gizli veri, başka bir veri içerisine fark edilmeyecek şekilde gizlenir. Örneğin gizli mesaj, imge dosyasının içerisine gizlenir ve gizli mesajı barındıran imge, orijinalinden ayırt edilemez. Bu şekilde haberleşen iki taraf arasında veri iletişimi son derece gizli ve güvenli olmaktadır [1]. Veri gizleme tekniği ile iki kişi haberleşirken üçüncü kişinin bu haberleşmeyi farketmesi oldukça güçtür. Veri gizleme tekniklerinden en çok kullanılan ve en basiti olan en düşük bite veri gizleme tekniği, birçok çalışmada kullanılmıştır. Bu teknik LSB (Least Significant Bit - En Önemsiz Bit) olarak adlandırılır [2]. LSB tekniğinin pek çok zayıflıkları vardır. Örneğin taşıyıcı dosyaya gürültü eklenmesi ile LSB tekniği ile gizlenen veri yok edilebilir [2]. Veri gizleme tekniklerinin tersine çalışan gizli veri analiz çalışmalarına da steganaliz adı verilmektedir [1]. Veri gizleme teknikleri kötü niyetli kişiler tarafından kullanımı büyük zararlarla sonuçlanabilir. Bunun önüne geçilebilmesi için steganaliz çalışmaları ile taşıyıcı ortamdaki gizli bilginin varlığı tespit edilebilir. Steganaliz çalışmalarının temelinde, veri gizleme işlemlerinin bıraktığı izleri sürmek vardır. Her veri gizleme işlemi birtakım parmak izleri bırakmaktadır. Bunu tesbit eden steganaliz çalışmaları taşıyıcı ortamdaki gizli verinin varlığını sezebilir [3].

Şekil 1'de steganografi ile veri gizleme ve veri çıkarma işleminin genel diyagramı verilmiştir. Şekil 1'e göre gizli imge, orijinal imgeye gizleme yöntemiyle gizlenerek taşıyıcı imge elde edilir. Taşıyıcı imge, orijinal imgenin formatını ve görünüm özelliklerini birebir barındırır. Taşıyıcı imgede yapılan değişiklikler insan gözü tarafından fark edilemez. Taşıyıcı imge karşı tarafa iletildikten sonra, karşı taraf çıkartma yöntemini kullanarak gizli imgeyi elde eder. Bu şekilde masum görünen bir imge ile gizli bilgiler taşınabilir.

*Yazışılan yazar/Corresponding author

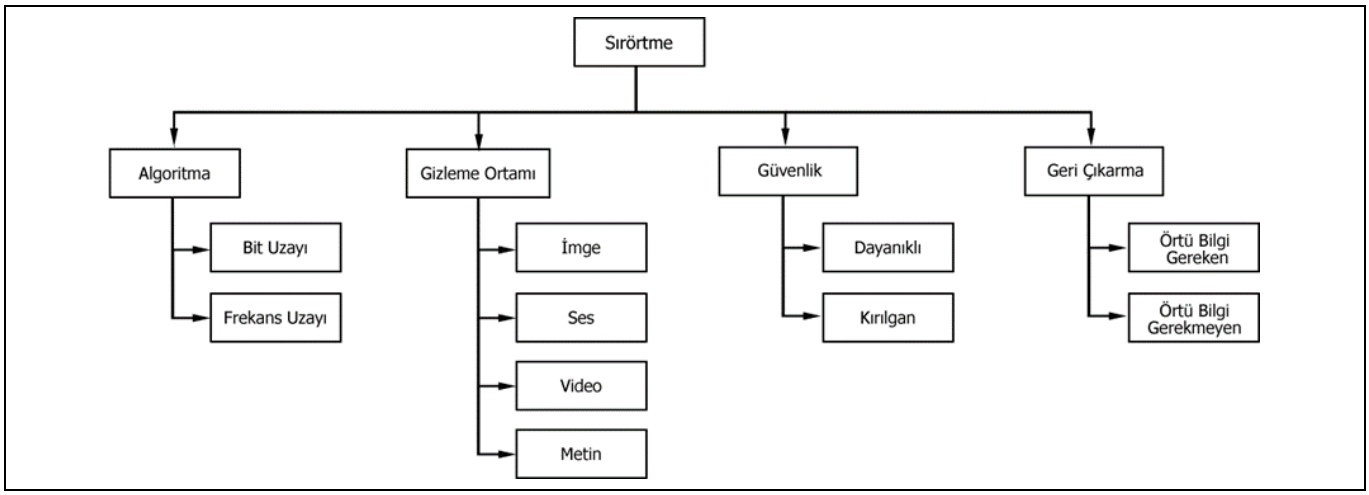
Şekil 1. Steganografide veri gizleme ve veri çıkarma işleminin genel diyagramı.

Figure 1. General diagram of data hiding and data extraction in steganography.

Steganografi, kullanılan algoritmaya göre, gizli mesajın gizlendiği veri ortamına göre ve güvenliğe göre üç ana başlık altında sınıflandırılabilir [4]. Şekil 2'de steganografi sınıflandırma şeması verilmiştir. Steganografi yöntemlerinde algoritma türlerine göre bit ve frekans uzayı, gizleme ortamlarına göre imge, ses, video ve metin, güvenlik özelliklerine göre dayanıklı ve kırılgan, geri çıkarma yöntemlerine göre ise örtü bilgi gerektiren ve gerektirmeyen olmak üzere iki yöntem bulunmaktadır [1]. Önerilen YKKG yönteminde algoritma türü olarak bit uzayı, gizleme ortamı olarak imge, güvenlik özelliklerine göre dayanıklı ve geri çıkarma yöntemlerine göre ise örtü bilgi gerektirmeyen steganografi yöntemi kullanılmıştır. Geri çıkarma yönteminde örtü bilgi gerektirmemesi tek bir imge dosyası ile haberleşmenin yapılmasına olanak sağlamaktadır.

Önerilen yöntem görüntü damgalama yöntemi değildir. Önerilen yöntem veri gizleme diğer adıyla steganografi yöntemidir. Steganografi, damgalama yöntemlerinden tamamen farklıdır. Sayısal damgalama yöntemlerinde genelde görülmeyen (görüntü, video ve metin için) veya duyulmayan (ses için) gizli bir işaret sayısal çoklu ortam verilerine eklenir ve damga adı verilen söz konusu işaret, verinin bütün kullanım ömrü boyunca mevcuttur [5]. Damgalama yöntemi içerikte telif hakkı koruması, veri doğrulaması, veri sahiplik kontrolü gibi amaçlar ile yapılırken, steganografi yöntemi gizli içeriğin üçüncü şahıslar tarafından fark edilmeden seçilen cover (taşıyıcı) ortama gizlenerek karşı tarafa iletilmesi amacını taşımaktadır.

Bit uzayı yöntemlerinden en yaygın kullanılanı diğer yöntemlere göre basit olmasından dolayı LSB en önemsiz bite gizleme yöntemidir [1]. Bu yöntemde taşıyıcı imge ve gizlenecek bilgi ikili sisteme çevrilerek gizleme işlemi yapılır. Taşıyıcı imgedeki her bir baytlık verinin en önemsiz biti değiştirildiğinde, taşıyıcı imgede insan gözü ile algılanacak bir değişiklik oluşmaz. Bu durumdan yararlanarak tasarlanan LSB yönteminde son bitler veri gizlemek amacıyla kullanılır. Bit uzayını kullanan bir diğer yöntem ise eşleştirme yöntemidir [1]. Eşleştirme yönteminde LSB yönteminde olduğu gibi son bitler doğrudan değiştirilmez. Bunun yerine en önemsiz bit, gizlenecek bit ile aynı değere sahip değilse ilgili bayt 1 arttırılır veya 1 azaltılır, aynı değerde ise o bit üzerinde değişiklik yapılmaz. Eşleştirme yöntemi ilk kez Sharp [6] tarafından önerilmiştir. Sharp [6] önerdiği yöntemde sözde-rastgele dizileri oluşturmak için stego anahtarları kullanan bir şema geliştirmiştir. Sharp [6] stego anahtar dizi kullanarak gizli içeriği şifreleyerek gizlemiştir. Mielikainen [7] Sharp'ın [6] önerdiği yöntemi geliştirerek gizleme işlemini, fonksiyonun verdiği sonuca göre yapmaktadır. Yönteme göre taşıyıcı imge piksel çiftlerine ayrılmıştır. Gizli bilgi, iki bitlik guruplar halinde gizlenmektedir. Gizlenecek ilk bit ilk pikselin son bitine doğrudan gizlenmektedir. İkinci bit ise piksel çiftinin son bitlerinin gizleme fonksiyonun verdiği sonuçla elde edilir. Fonksiyon sonucunun ikinci gizleme bitini verebilmesi için piksel çiftlerinden birisi 1 arttırılır veya 1 azaltılır. Mielikainen'in [7] yöntemi LSB yöntemine göre taşıyıcı imgede daha az değişiklik yapmaktadır. Bunun yanında LSB yöntemiyle aynı kapasitede veri gizlemektedir. Chan [8], Mielikainen'in [7] yaptığı çalışmayı geliştirerek yeni bir yöntem önermiştir. Yöntem, imgedeki ardışık iki pikselden ilk pikselin LSB biti ile ikinci pikselin LSB bitinden bir önceki bitine XOR yöntemi uygulayarak veri gizlemektedir.



Şekil 2. Steganografinin sınıflandırılması [4].

Figure 2. Classification of steganography [4]

Eşleştirme yöntemini kullananan Tian [9] taşıyıcı imgede düşük bozulumlu, yüksek kapasiteli ve tersinir veri gizleme yöntemi önermiştir. Yöntemin kapasitesi yüksek olduğu için imgeye yüksek oranda veri gizleyebilmektedir. Tersinir olduğu için de orjinal imgeye ihtiyaç duymadan stego imgeden gizli veri geri çıkartılabilir. Tian [9], önerdiği yöntemde taşıyıcı imgenin iki pikseli arasındaki farkı iki kat genişleterek oluşan bölgeye veri gizlemiştir. Alatlar [10], Tian'ın önerdiği yöntemi geliştirmiş ve dört piksel arasındaki farkı iki kat genişleterek 3-bitlik veriyi oluşan bölgeye gizlemiştir. Chang ve diğ.[11] yaptıkları çalışmalarında taşıyıcı imgeyi iki kez oluşturdular. Oluşan iki imgeye modül matrisi ve değişiklik yönünü kullanarak veri gizlediler. İki imge olduğu için yüksek veri kapasitesi sundular. Lu ve diğ. [12] Chang'in [9] önerdiği yöntemi geliştirerek alternatif bir yöntem önerdiler. Önerilen yöntem, kamuflaj pikselleri ile taşıyıcı imgelerde yüksek görüntü kalitesinin yanında yüksek veri gizleme kapasitesi sunmaktadır. Ker [13] yaptığı çalışmada 2/3 verimli gizleme yöntemi önermiştir. Bu yöntem iki bit veri gizlemek için üç piksel kullanır. İki pikselin son biti veri gizleme amacında kullanılırken son piksel gizlenen verinin aynı şekildemi yoksa tümleyeninin mi gizlendiğini gösterir. Eğer gizlenecek iki bit, tümleyeni alınarak gizlendiğinde taşıyıcı imgenin son bitlerinde daha az değişiklik yapacaksa tümleyen olarak, değilse değişiklik yapılmadan gizlenir. Böylece son bitlerde daha az değişiklik yaparak veri gizlemeye çalışır. Wu ve Tsai [14] piksel farkı ile veri gizleme yöntemini önermişlerdir. Bu yöntemde taşıyıcı imgedeki ardışık pikseller çakışmayacak şekilde üst üste getirilerek piksel değerlerinin farkları hesaplanır. Olabilecek fark değerleri farklı sınıflarla temsil edilir. Fark değerlerinin yerine yeni bir veri gizlenerek gizleme işlemi sağlanır. Wang ve diğ. [15] Wu ve Tsai'nin [14] yaptığı çalışmayı geliştirerek yeni bir yöntem önermişlerdir. Yeni yöntem, iki piksel arasındaki farkın modül fonksiyonu sonucunu kullanarak veri gizleme esasına göre çalışmaktadır. Fridrich ve Soukal [16] hamming matrisini kullanarak veri gizleme yöntemi önermişlerdir. Yöntem taşıyıcı imgede diğer yöntemlere göre daha az değişiklik yapmaktadır. Ayrıca yöntemin yüksek veri kapasitesi ile veri gizleme özelliği de vardır. Kurtuldu ve Arıca [17] imge kareleri yöntemi adını verdikleri çalışmalarında yeni bir veri gizleme yöntemi önermişlerdir. Önerdikleri yöntemde, taşıyıcı imge bloklara bölünür. Gizlenecek mesaj verisini blok piksellerinin son bitlerinin dizilimlerine bakarak mesaj verisine en yakın dizilime sahip piksel gurubuna gizlenmektedir. Bu yöntemde taşıyıcı imgeye gizlenen bilginin kapasitesi düşüktür. Wu ve diğ. [18] çalışmalarında stego imgeyi iki kez oluşturarak ilk stego imgeye veriyi, ikinci stego imgeye ise gizlenen verinin referans bilgilerini gizlediler. Wu ve diğ. [18] önerdikleri yöntemde stego imgelerden gizli veri, Diffie-Hellman (DH) anahtarı denilen bir yöntem kullanarak çıkartılır. DH anahtar değişimi metodu karşılıklı iki tarafın ortaklaşa güvensiz medya üzerinden ortak gizli anahtar elde etmelerine olanak sağlar. Bu anahtar daha sonra bir simetrik anahtar şifre kullanarak sonraki güvenli olmayan kanaldan iletişimi şifrelemek için kullanılabilir [19]. Simetrik anahtar algoritmaları aynı ya da benzer kriptografik şifreleri kullanarak hem şifreleme hem deşifreleme yapan bir kriptografik algoritma grubudur. Şifreler ya birebir aynı ya da basit bir yöntemle birbirine dönüştürülebilir olmalıdır [20]. Huang ve diğ. [21] kenar uyarlamalı bitişik piksel çifti eşleştirme yöntemi ile veri gizlemişlerdir. Önerilen yöntemde bitişik piksel çifti seçiminde yeni bir rasgele seçim yöntemi kullandılar. Sabeti ve diğ. [22] yaptıkları çalışmada, imgedeki veri gizleme işlemi için güvenli bölgeyi belirlemek için karmaşıklık ölçütü kullanmışlardır. Saldırılara karşı geleneksel eşleştirme yöntemlerine göre daha güvenli olduğu göstermişlerdir. Jain ve Kumar [23] kayıpsız veri sıkıştırması ile imge içine yüksek kapasiteli veri gizleme yöntemini önermişlerdir. Jain ve Kumar yaptıkları çalışmalarında gizli veriyi kayıpsız veri sıkıştırmasıyla gizlemişlerdir. Atıcı ve Sağıroğlu [24] steganografi tabanlı yeni bir klasör kilitleme yaklaşımı önermişlerdir. Önerdikleri yöntemde windows ortamında seçilen klasör ve içindeki dosyalar steganografi yöntemi ile kilitlenerek güvenceye alınır.

Frekans uzayı yöntemini kullanan çalışmalarda mevcuttur. Emek [25] frekans uzayı yöntemini kullanarak geliştirdiği sabit görüntüler ve video hareketleri için ayrık dalgacık dönüşümlü ayrık kosinus dönüşümlü sayısal damgalama yöntemini önermiştir. Ruanaidh ve diğ. [26] dijital görüntüler için telif hakkının korumasının sağlayacak gizleme yöntemi tanıtmışlardır. Podilchuk ve Zeng [27] görüntü sıkıştırma bağlamında geliştirilen görsel modelleri kullanmaya dayalı dijital görüntüler için telif hakkının korumasının sağlayacak bir gizleme yöntemi önermişlerdir. Hernández ve diğ. [28] durağan imgeler için frekans uzayında ayrık kosinus dönüşümlü damgalama yöntemlerini analiz etmişlerdir. Hartung ve Girod [29] sıkıştırılmış ve sıkıştırılmamış videolar için damgalama yöntemi sunmuşlardır. Bhatagar and Raman [30] ayrık dalgacık dönüşümlü ve tekil değer ayrıştırma bazlı telif hakkının korumasının sağlayacak yeni bir yarı-kör referans gizleme yöntemi şeması sunmuşlardır. Zeng ve diğ. [31] yaygın spekturum tekniği ile gizlenen veriler için iki adet steganaliz yöntemi geliştirmişlerdir. Baby ve diğ. [32] ayrık dalgacık dönüşümü kullanılarak birden fazla renkli görüntüyü tek bir renkli görüntüye gizleyen yeni bir veri gizleme yöntemi önermişlerdir. Chen ve Lin [33] gizli mesajları frekans alanına yerleştiren yeni bir steganografi tekniği sunmuşlardır. Kamila ve arkadaşları ayrık dalgacık dönüşümlü renkli imgeler için steganografi tekniği tanıtmışlardır [34]. Bhattacharyya ve Kim görüntünün kimliğini doğrulamak için ayrık fourier dönüşümüne dayalı yeni bir yöntem geliştirmişlerdir [35]. Chen diferansiyel faz kaydırmalı anahtarlama tekniğine dayanarak, taşıyıcı imge ile aynı boyutta imge gizleyen veri gizleme yöntemi sunmuşlardır [36]. Tuncer ve Avcı Göktürk alfabesi tabanlı görsel sır paylaşımı yöntemi ile yeni bir veri gizleme yöntemi önermişlerdir [37]. Chen ve diğ. [38] JPEG yapısını mikro ölçekte kullanarak JPEG steganografisinin maliyet fonksiyonunu iyileştirmek için bir yöntem önermişlerdir. Sarmah ve Kulkarni [39] JPEG görüntü steganografisinde yüksek kapasiteli, hızlı ve güvenli bir yaklaşım önermişlerdir. Khosravi ve diğ. [40] pdf dosyaları için Huffman sıkıştırma algoritmasını kullanarak metin tabanlı bir gizleme yöntemi sunmuşlardır.

Ayrıca Doğan [41] yaptığı çalışmada genetik algoritmaya dayalı veri gizleme tekniğini geliştirmek için kaotik haritalar kullanılmıştır. Yine Doğan [42] bir başka çalışmasında grafik blok komşuluk derece tabanlı tersinir veri gizleme şeması önermiştir. Tuncer ve diğ. [43] yaptıkları çalışmalarında ikili imgeler için mayın tarlası oyunu tabanlı yeni bir veri gizleme algoritması önermişlerdir. Önerdikleri bu yeni yöntemde mayın tarlası oyunundan ilham almışlardır. Görüntü damgalama yöntemlerine örnek olabilecek çalışmada Tuncer [44] yüksek kapasite, yüksek görsel kalite, yüksek görüntü özgünlüğü yeteneği ve görüntü kurtarma özelliğine sahip yeni bir olasılıklı görüntü kimlik doğrulama yöntemi önermiştir. Bir

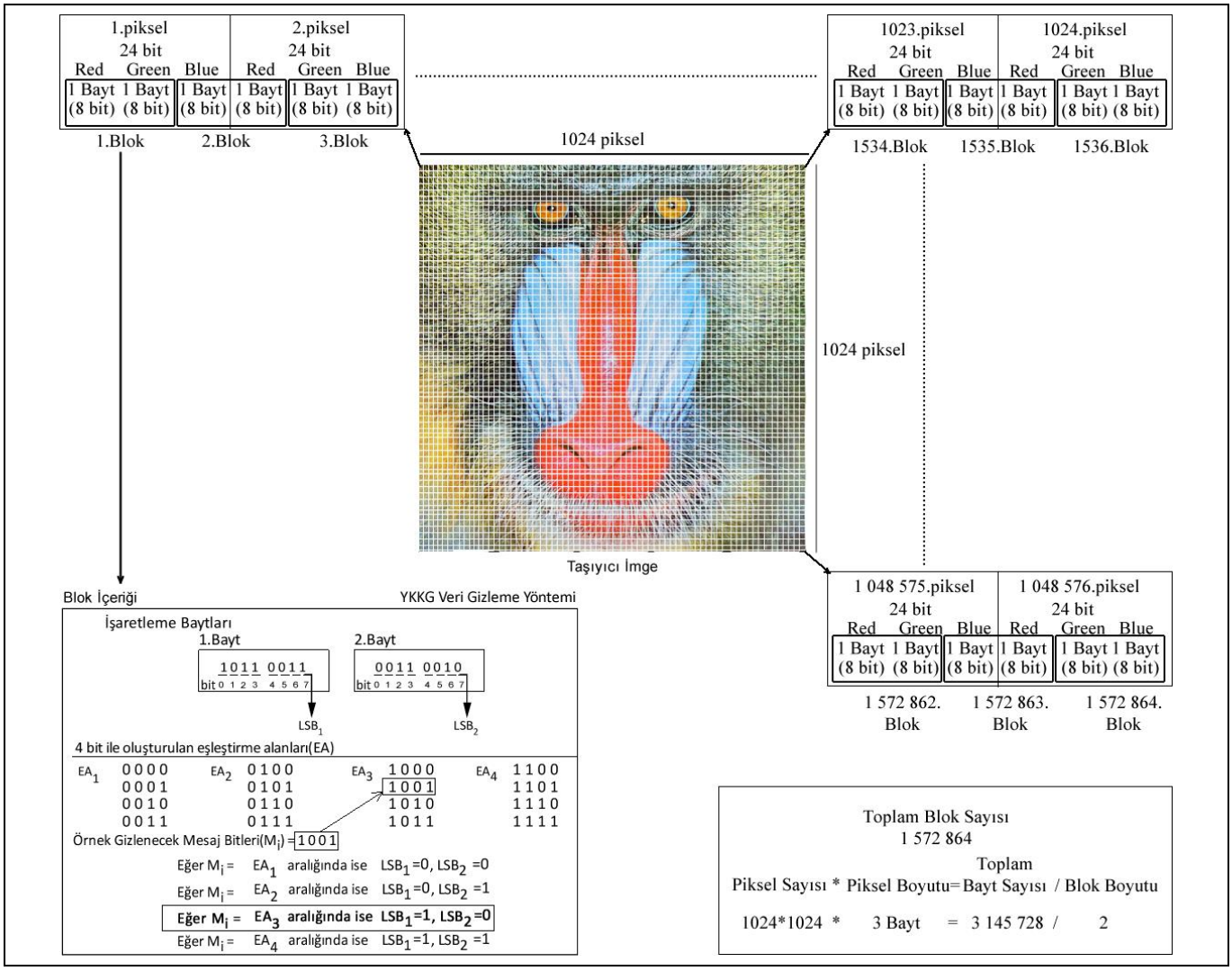
başka çalışmasında Tuncer [45] yerel ikili örüntü tabanlı veri gizleme algoritması önermiştir. Kıllıoğlu ve diğ. [46] yaptıkları çalışmalarında steganografi ve şifreleme kullanılarak çoklu biyometrik sistemlerle kimlik doğrulama için güvenli veri iletimi yöntemi önermişlerdir.

Bu çalışmada, 24-bit renkli imge içine 24-bit renkli imgeyi kayıplı gizleyen yüksek kapasiteli, düşük bozulumlu, kayıplı ve tersinir yeni bir veri gizleme yöntemi (YKKG) önerilmiştir. Önerilen yöntem veri gizleme kapasitesi olarak yüksek kapasite sunarken, taşıyıcı ortamdaki veri değişikliği için ise düşük bozulum oluşturmaktadır. Bölüm 2'de YKKG yöntemi, bölüm 3'de ise YKKG yönteminin başarım performansı değerlendirmek için deneysel sonuçlar verilmektedir.

2 Önerilen metod

YKKG yöntemi, 24-bit renkli imgeye 24-bit renkli imgeyi kayıplı gizlemektedir. Şekil 3'te önerilen YKKG yönteminin çalışma prensibi gösterilmiştir. Şekil 3'te görüldüğü gibi taşıyıcı imge 2 baytlık bloklara bölünerek gizleme işlemi yapılır. 24-bitlik bir imgedeki her bir pikselde Red-Green-Blue (Kırmızı (8bit)-Yeşil (8bit)-Mavi (8bit)) her bir renk kanalı için 8 bit olmak üzere 24 bit yer ayrılmıştır. Buna göre her bir renk kanalı 1 bayt her bir piksel ise 3 baytlık yer kaplar. Şekil 3'te 24-bitlik 1024x1024 boyutlarında imge verilmiştir. Bu imgede 1024 * 1024 toplam 1 048 576 adet piksel bulunmaktadır. Şekil 3'te piksellerin içeriği gösterilmiştir. 1.pikselde 3 bayt bulunmaktadır. YKKG yönteminde oluşturulan bloklar piksellere göre değil baytlara göredir. İmgedeki toplam bayt sayısı 2-baytlık gruplar halinde bloklara ayrılır. Buna göre 1.pikselin ilk 2 baytı 1.bloğu oluşturmaktadır. 1. pikselin son baytı ile 2. pikselin ilk baytı 2. bloğu, 2. pikeselin son iki baytı ise 2. bloğu oluşturmaktadır.

Buna göre 1024x1024 piksellik 24-bit bir imgede 1024*1024*3bayt=3 145 728 bayt bulunmaktadır. Toplam bayt sayısını blok boyutuna (2 bayt) böldüğümüzde ise toplam oluşan blok sayısını 1 572 864 blok elde edilir. YKKG yönteminde her bir bloğa 4-bit veri gizlenmektedir. Önerilen yöntem gizlenecek 24-bitlik imgeyi, 4-bitlik parçalara böler ve her parçayı 2-bitlik gizleme koduna indirgeyerek gizler. Bu şekilde 4-bitlik parça 2-bite indirgendiği için yöntem kayıplı gizleme yapmaktadır. Gizlenen imgenin her bir renk kanalına için piksel bazında hata büyüklüğü $(00100010)_2$=34'tür.



Şekil 3. YKKG yönteminin çalışma prensibi.

Figure 3. Working principle of YKKG method.

Hatanın düzgün dağıldığını düşünecek olursak her bir kanal için ortalama hata 34/2=17'dir. 2-bitlik gizleme kodları 2-baytlık bloklara gizlenir. Geri çıkarma işleminde ise yöntem, 24-bit imge gizlenmiş 24-bit stego imgeden, sırasıyla 2-baytlık bloklardaki 2-bitlik gizleme kodlarını kullanarak 4-bitlik parçalar elde eder ve parçalar birleştirilerek 24-bitlik gizli imge tersinir olarak geri elde edilir. Şekil 3'teki 24-bitlik imgeye 1 572 864 * 4-bit = 6 291 456-bit / 8 = 786 432 bayt veri gizlenebilir. Şekil 3'te gizleme yapılan blokların içeriği büyütülerek gösterilmiştir. Her blok 2 bayt boyutundadır. Her bir baytta 8 bit veri bulunmaktadır. 8-bit verinin soldan 8.biti (bit_7) en önemsiz bitdir. Bu bitte yapılacak değişiklik baytın sayısal değerini 1 azaltır veya 1 arttırır. Örneğin baytın değeri 234 $(1110\ 1010)_2$ olsun. Soldan 8.bit olan LSB biti 0'dır. LSB bitinin 1 yapılmasıyla baytın değeri $(1110\ 1011)_2 = 235$ olur. Bu küçük değişiklik pikselin renk tonunu değiştirmez. Bu şekilde LSB yöntemi ile farkedilmeyecek şekilde veri gizlenebilir. Şekil 3'te görüldüğü gibi bloktaki 1. baytın en önemsiz biti LSB_1 2.baytın en önemsiz biti ise LSB_2 olarak isimlendirilmiş ve bu bitler işaretleme yapmak için kullanılır. YKKG sisteminde Şekil 3'te blok içeriğinde verilen 4 adet eşleştirme alanı bulunmaktadır.

Önerilen YKKG yöntemi YKKG yönteminde taşıyıcı imge dosyası 2 baytlık bloklara bölünerek, gizleme işlemi ardışık olarak yapılır. Gizleme işleminde, her bir bloktaki 2 baytın son bitleri birlikte kullanılarak EA_1, EA_2, EA_3 ve EA_4 olmak üzere dört adet eşleştirme alanı oluşturulur. Her bir eşleştirme alanı 4 bitliktir ve bir önceki eşleştirme alanının değerlerinin sayı ardışığıdır. Gizli mesajda eşleştirme alanlarına uygun olacak şekilde 4-bit parçalara bölünerek gizlenir. Dört adet eşleştirme alanı ile onluk sayı sistemine göre $0(0000)_2$ - $15(1111)_2$ arasındaki tüm sayı değerleri temsil edilmektedir. Gizlenecek 4-bit mesaj parçası ile $0(0000)_2$-$15(1111)_2$ arasında değerler oluştuğundan gizlenecek veri parçası eşleştirme alanlarından birisi ile mutlaka eşleşir. Her eşleştirme alanı 2-bit ile kodlanarak ve 4 bitlik mesaj verisi 2-bit ile temsil edilmiştir. Buna göre eşleştirme alanları ve gizleme işlemindeki kod karşılık değerleri, ikili ve onluk sayı sistemi ile birlikte gösterilerek verilmiştir. 1. eşleştirme alanındaki $0(0000)_2$-$3(0011)_2$ arasındaki değerler $0(00)_2$ ile; 2. eşleşme alanındaki $4(0100)_2$ - $7(0111)_2$ arasındaki değerler $1(01)_2$ ile; 3.eşleşme alanındaki $8(1000)_2$ - $11(1011)_2$ arasındaki değerler $2(10)_2$ ile ve 4.eşleşme alanındaki $12(1100)_2$-$15(1111)_2$ arasındaki değerler ise $3(11)_2$ ile kodlanarak gizlenir.

Önerilen YKKG yönteminin matematiksel gösterimi Eş. 1'de verilmiştir. Eş. 1'de SB stego bloğu (SB), C_i taşıyıcı imgenin i. baytını, LSB(C_i) i.baytın en önemsiz bitini, EA eşleştirme alanlarını, m_i i.bayta gizlenecek 4-bitlik mesaj parçasının onluk sayı sistemindeki, M_i i.bayta gizlenecek 4-bitlik mesaj parçasının ikilik sayı sistemindeki karşılığını göstermektedir. Eş. 1'e göre $EA_1 = \{0,1,2,3\}$, $EA_2 = \{4,5,6,7\}$, $EA_3 = \{8,9,10,11\}$ ve $EA_4 = \{12,13,14,15\}$ değerlerini alır. Ayrıca Eş. 1'de stego blokta (SB) 2-bit LSB biti ile 4-bit verinin nasıl temsil edildiği verilmiştir. Eş. 1'de eşleştirme alanlarının aralıkları onluk sayı sistemi ile gösterilmiştir. Gizlenecek 4-bit mesaj parçası onluk sayı sistemine çevrilerek eşleştirme alanlarından hangisinin aralığına giriyorsa $LSB(C_i)$ ve $LSB(C_{i+1})$ bitleri Eş. 1'de gösterildiği gibi değiştirilerek gizleme işlemi yapılır.

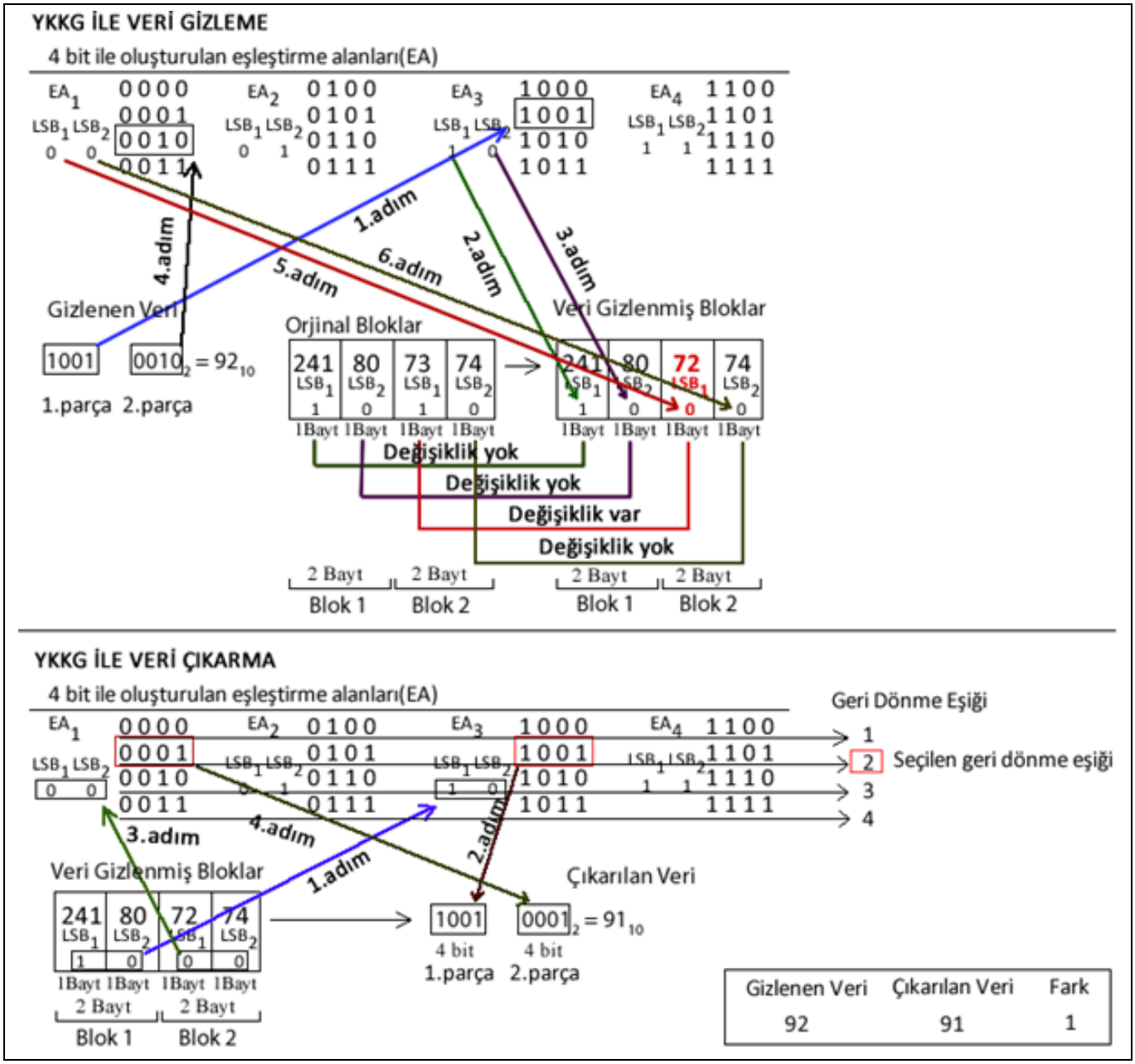
$$SB = \begin{cases} LSB(C_i) = 0, LSB(C_{i+1}) = 0 \;\; EA_1 = \{ m_i \mid 0 \le m_i \le 3, \quad M_i \in Z\} \\ LSB(C_i) = 0, LSB(C_{i+1}) = 1 \;\; EA_2 = \{ m_i \mid 4 \le m_i \le 7, \quad M_i \in Z\} \\ LSB(C_i) = 1, LSB(C_{i+1}) = 0 \;\; EA_3 = \{ m_i \mid 8 \le m_i \le 11, \quad M_i \in Z\} \\ LSB(C_i) = 1, LSB(C_{i+1}) = 1 \;\; EA_4 = \{ m_i \mid 12 \le m_i \le 15, \quad M_i \in Z\} \end{cases} \quad (1)$$

YKKG yönteminde Eş. 1'e göre bir imge 2-bayt boyutunda Şekil 3'teki gibi n adet bloğa bölünür ve her bloğun (SB_i, $i = 1 \ldots n$), iki adet LSB biti bulunur. Bloklardaki bu 2-bit veri işaretleme amacıyla kullanılarak her bloğa 4-bit mesaj parçası m_i, kayıplı olarak gizlenir. Buna göre bir imgeye $n * 4$ bit uzunluğunda mesaj gizlenebilir. Bunu bir örnekle gösterecek olursak; 600KB bir imgede 2 baytlık 307 200 blok oluşur. Toplamda 307 200 * 4=1 228 800 bit veri gizlenebilir. 1 228 800 / 8=153 600 bayt, 153 600 / 1024=150KB veri gizlenebilir. Geleneksel LSB yönteminde 600KB bir imgeye 8'de 1 oranında 600 / 8 = 75 KB veri gizlenebilir. Buna göre YKKG yöntemi geleneksel LSB yönteminin 2 katı oranında veri gizlemektedir.

Gizlenecek 4-bit veri M_i, dört eşleşme alanından hangisinin aralığına giriyorsa ona göre taşıyıcı imgede 2-bit işaretleme yapılır. Şekil 3'te dört adet eşleştirme alanı ve içerdiği değerler gösterilmektedir. Şekil 3'teki örnek gizlenecek veri $M_i = (1001)_2$ EA_3'ün eşleşme aralığında olduğu için LSB_1 $(1)_2$ ile LSB_2 ise $(0)_2$ ile değiştirilerek gizli veri işaretlenir. Böylece gizli verinin eşleştirme alanının kod karşılık değeri bloğa gizlenmiş olur. Geleneksel LSB veri gizleme yönteminde LSB bitleri ya 1 yada 0 olarak değiştirildiğinden tüm veri gizleme işlemi sonucunda taşıyıcı dosyanın LSB bitleri ortalama %50 oranında değişmektedir. YKKG yönteminde ise, geleneksel LSB yöntemi ile aynı oranda veri gizlemesi yapıldığında 4 bitlik veri 2-bit olarak gizlendiği için taşıyıcı dosyanın LSB bitleri ortalama %25 oranında değiştirmektedir. Ayrıca YKKG yönteminin kapasitesi, gizlenen veriyi kayıplı sıkıştırma yöntemiyle gizlediği için geleneksel LSB yöntemlerininin 2 katıdır.

Şekil 4'te YKKG yöntemiyle veri gizleme ve çıkarma işlemleri örneklendirilerek gösterilmektedir. Şekil 4'teki sayısal değerler YKKG yöntemini örnekle gösterebilmek için oluşturulmuştur. Her biri 2-bayt boyutundaki Blok 1 ve Blok 2 adlı iki bloğa $(1001\ 0010)_2$ 8 bitlik veri YKKG yöntemi ile gizlenmektedir. Gizlenecek veri 4 bitlik parçalara bölünerek iki parça olarak gizlenmektedir. Gizli verinin 1.parçası $(1001)_2$, Şekil 4'te 1.adımdaki ok ile gösterildiği gibi EA_3'ün eşleştirme aralığında olduğu için EA_3'ü işaret eden $(10)_2$ gizleme kodu $(10)_2$ 2.adımdaki ok ile gösterilen Blok 1'in LSB_1 biti 1, 3.adımda ok ile gösterildiği gibi Blok 1'in LSB_2 biti ise 0 ile değiştirilerek gizlenir. Blok 1'in LSB_1 ve LSB_2 bitleri zaten $(10)_2$ olduğu için veri gizleme sonucunda herhangi bir değişiklik oluşmamıştır. Gizli verinin 2. parçası $(0010)_2$, Şekil 4'te 4. adımdaki ok ile gösterildiği gibi EA_1'in eşleştirme aralığıda olduğu için EA_1'i işaret eden gizleme kodu $(00)_2$, 5. adımdaki ok ile gösterilen Blok 2'in LSB_1 biti 0, 6. adımda ok ile gösterildiği gibi Blok 2'in LSB_2 biti de 0 ile değiştirilerek gizlenir. Blok 2'deki 73 ve 74 değerlerinin LSB_1 ve LSB_2 ile $(10)_2$ değerine sahiptir. Gizleme işleminden sonra Blok 2'nin ilk baytındaki 73 değeri 72 olarak değişmiş ve Şekil 4'te kırmızı ile gösterilmiştir.

Şekil 4'te gizli verinin çıkarma işlemi de gösterilmiştir. YKKG yönteminin veri çıkarma işleminde, gizleme kodunun gösterdiği eşleştirme alanının hangi değerinin geri döneceğini geri dönme eşik değeri belirler. Eşleştirme alanlarında dört adet veri bulunmaktadır. Geri dönen değerin belirlenmesinde tüm veri çıkarma işlemi boyunca bu dört adet veriden geri dönme eşiğine karşılık gelen değer geri çıkarılır. Şekil 4'te veri gizlenmiş Blok 1'in LSB_1 ve LSB_2 bitleri $(10)_2$ değerine sahiptir. Bu bilgi gizleme kodudur. Gizleme kodu 1.adımdaki ok ile gösterildiği gibi EA_3 eşleştirme alanını göstermektedir.



Şekil 4. YKKG yöntemi ile örnek veri gizleme ve çıkarma

Figure 4. Sample data hiding and extraction with YKKG method

Şekil 3'teki örnekte seçilen geri dönme eşiği 2 olduğu için eşleştirme alanının yukarıdan ağaşıya 2.sıradaki $(1001)_2$ verisi geri çıkartılır. 2. adımdaki ok ile gösterilen çıkarılan verinin 1. parçası bu şekilde elde edilir. Veri gizlenmiş Blok 2'nin LSB_1 ve LSB_2 bitleri $(00)_2$ değerine sahiptir. Gizleme kodu 3.adımdaki ok ile gösterildiği gibi EA_1 eşleştirme alanını göstermektedir. Eşleştirme alanının yine yukarıdan 2.sıradaki $(0001)_2$ verisi geri çıkartılarak geri çıkarılan verinin 2. parçası elde edilir. Sonuç olarak Şekil 4'te görüldüğü gibi onluk sayı sisteminde 92 verisi gizlenmiş ve 1 fark ile 91 verisi geri elde edilmiştir. YKKG yöntemi kayıplı veri gizlemektedir. Yöntemin bu özelliğinde dolayı sadece imge veri gizleyebilmektedir. Geri çıkarma işlemi sonucunda oluşan kayıplar geri elde edilen imgenin kalitesinde bozulma oluştursa da anlaşılmayacak derece imge bozulmamaktadır. YKKG yönteminin en önemli özelliği, kayıplı veri gizlediğinden Şekil 4'te de görüldüğü gibi 8-bit veri gizlenmesine rağmen sadece 1-bit veri değişikliği oluşturmasıdır. Ayrıca önerilen YKKG yöntemi geleneksel LSB yöntemlerine göre 2 kat veri gizlemektedir.

Tablo 1'de YKKG yönteminde gizlenen verinin geri dönme eşiğine göre geri elde edilen verideki değişim tablosu verilmiştir. Tablo 1'de gizlenen veri onluk sayı sisteminde, gizlenen veriye karşılık gelen gizleme kodu ikili sayı sisteminde ve geri dönme eşiğine göre geri elde edilen değer onluk sayı sisteminde gösterilmektedir.

Tablo 1. YKKG yönteminde gizlenen veri ve geri dönme eşiğine göre geri elde edilmesi.

Table 1. Data hidden in the YKKG method and extract according to the return threshold.

Gizlenen Veri	Gizleme Kodu	Geri Dönme Eşiği				Eşleştirme Alanı
		1	2	3	4	
		Geri Elde Edilen Veri				
0	00_2	0	1	2	3	EA_1
1	00_2	0	1	2	3	EA_1
2	00_2	0	1	2	3	EA_1
3	00_2	0	1	2	3	EA_1
4	01_2	4	5	6	7	EA_2
5	01_2	4	5	6	7	EA_2
6	01_2	4	5	6	7	EA_2
7	01_2	4	5	6	7	EA_2
8	10_2	8	9	10	11	EA_3
9	10_2	8	9	10	11	EA_3
10	10_2	8	9	10	11	EA_3
11	10_2	8	9	10	11	EA_3
12	11_2	12	13	14	15	EA_4
13	11_2	12	13	14	15	EA_4
14	11_2	12	13	14	15	EA_4
15	11_2	12	13	14	15	EA_4

1 ile 4 arasında dört adet geri dönme eşiği ve dört adet eşleştirme alanı bulunmaktadır. $(00)_2, (01)_2, (10)_2$ ve $(11)_2$ dört adet gizleme kodu sırasıyla EA_1, EA_2, EA_3 ve EA_4 eşleştirme alanlarını göstermektedir. Tablo 1'de, önerilen YKKG yöntemiyle gizlenen verinin, geri dönme eşiğine göre hangi veriyi geri elde edeceği göserilmektedir. Her eşleştirme alanında dört adet veri bulunmaktadır. Gizli veri geri elde edilirken öncelikle gizleme kodunun gösterdiği eşleştirme alanı bulunur. İkinci aşamada ise geri dönme eşiğinin eşleştirme alanındaki dört veriden hangisinin geriye döneceği seçilerek gizli veri geri elde edilir. Örneğin onluk sistemde 8-11 arası tüm değerleri gizlemek için $(10)_2$ gizleme kodu kullanılır. $(10)_2$ gizleme kodu, EA_3 eşleştirme alanını gösterir. Geri dönme işleminde, EA_3 geri dönme eşiğine göre belirlenen eşleştirme alanının değeri geri elde edilir. Örneğin 9 değeri geri çıkartılırken 1 geri dönme eşiğine göre 8, 2 geri dönme eşiğine göre 9, 3 geri dönme eşiğine göre 10 ve 4 geri dönme eşiğine göre ise 11 olarak geri elde edilir.

YKKG yönteminin veri gizleme işlemi Algoritma 1'de verilmiştir. Veri gizleme yönteminde *Tasiyici_Veri* değişkeni örtü imge verisini, Mesaj değişkeni gizlenecek veriyi, tasiyici_ikili örtü imgenin verisinin ikilik sayı sistemine dönüştürülmüş halini, *mesaj_ikili* değişkeni gizlenecek mesaj verisinin ikilik sayı sistemine dönüştürülmüş halini, *max_blok_sayi* değişkeni örtü imgedeki veri gizlenebilecek maksimum blok sayısını, *gizlenecek* değişken sıradaki gizlenecek 4-bitlik mesaj parçasını, *i* değişkeni örtü imgedeki gizleme yapılan sıradaki bayt indeksini, *pos* değişkeni gizlenecek gizli mesajın sıradaki bayt indeksini, *bit* değişeni gizli mesajın sıradaki baytının sıradaki 4 bitlik kısmını temsil etmektedir. Örneğin *bit* değişkeni 0 olduğunda gizlenecek mesajın sıradaki baytının ilk 4 biti alınırken *bit* değişkeni 4 olduğunda ise ikinci 4-bit alınarak bir baytlık veri sıralı olarak gizlenir. Bu şekilde baytlardaki bilgiler anlam bozukluğu olmadan sıralı olarak gizlenir. Aksi durumda bayt bilgisinin anlamı bozulursa gizlenen mesajın anlam bütünlüğü kaybolur. YKKG veri gizleme yöntemi *Tasiyici_Veri* ve *Mesaj* adlı iki parametre alarak çalışır. İlk olarak *Tasiyici_Veri* ve *Mesaj* ikili sayı sistemine dönüştürülür. Ardından maksimum blok sayısı hesaplanır ve tüm blokları sırayla işleme alacak şekilde döngü kurulur. İlk 4-bit mesaj parçası alınarak işleme başlanır. Gizlenecek 4-bit mesaj parçası onluk sayı sistemine çevrilir. Onluk sayı 0 ila 3 arasında ise geri çıkarma işlemi için bloktaki son bitler $(00)_2$; 4 ila 7 arasında ise bloktaki son bitler $(01)_2$; 8 ila 11 arasında ise bloktaki son bitler $(10)_2$; 12 ila 15 arasında ise bloktaki son bitler $(11)_2$ gizleme kodu ile değiştirilir. Bu şekilde gizlenecek bitler 2-bit'e indirgenerek taşıyıcı imgede daha az değişiklikle gizlenir. Bu sayede gizlenecek mesaj 4-bit parçalar halinde eşleştirme alanları ile eşleştirilerek gizleme kodları taşıyıcı imgeye gömülür.

YKKG yönteminin veriyi geri çıkarma işlemi Algoritma 2'de verilmiştir. Veri çıkarma yönteminde Stego_Veri değişkeni veri gizlenmiş stego imge verisini, geri_donme_esik değişkeni geri dönecek eşleştirme alanının değerlerinden hangi sıradaki değerin geri döneceğini, stego_ikili stego imgenin verisinin ikilik sayı sistemine dönüştürülmüş halini, max_blok_sayi değişkeni örtü imgedeki veri gizlenebilecek maksimum blok sayısını, geri_donen_deger değişkeni dört eşleştirme alanının tüm değerlerini tutan bir diziyi, mesaj_boyut değişkeni gizlenmiş mesajın boyutunu, i değişkeni örtü imgedeki gizleme yapılan sıradaki bayt indeksini temsil etmektedir. YKKG veri geri çıkarma yöntemi Stego_Veri ve geri_donme_esik adlı iki parametre alarak çalışır. Stego verinin ilk 20 biti gizli mesaj bilgisinin boyutunu göstermektedir.

Gizli mesaj bilgisinin boyutu okunduktan sonra geri çıkarma işleminde de gizli mesajın boyutu kadar stego blok okunur. Veri çıkarma işleminde eşik değeri eşleştirme alanındaki değerlerden hangi değerin geri döneceğini belirleyen kriter olmaktadır. Dört eşleştirme alanıyla dört geri dönme eşik değerine göre 16 farklı durum oluşmaktadır. Bu değerler geri_donen_deger dizisine kaydedilmiştir. Seçilen geri dönme eşik değerine göre veri çıkarma işlemi o geri dönme eşik değerine karşılık gelen eşleştirme alanının değeri ile yapılır. Çıkarma işlemi, gizli verinin boyutu kadar dönen bir döngü ile tüm blokları baştan sona tarayarak devam eder. Her tarama

işleminde sıradaki bloğun son bitleri okunarak elde edilen gizleme kodu ve geri dönme eşik değerinin gösterdiği eşleştirme alanındaki değer ile gizli veri geri çıkartılır. Her stego bloktan 8-bit gizli veri parçası çıkarılır. Tüm bloklar okunduğunda gizli veri parçaları birleştirilerek gizli mesaj bilgisi elde edilir. YKKG yöntem imge içine imge gizlediği için gizli mesaj bilgisi imge formatında kaydedildilerek gizli imge geri çıkartılır.

Algoritma 1. YKKG Veri Gizleme Yöntemi.

```
procedure VERIGIZLE(Tasiyici_Veri,Mesaj)
  max_blok_sayi = sizeof(Tasiyici_Veri)/2
  tasiyici_ikili = convert to binary(Tasiyici_Veri)
  mesaj_ikili = convert to binary(Mesaj)
  mesaj_boyut = sizeof(Mesaj)
  i=1, pos=1, bit=0
  while i<max_blok_sayi do
    if pos >= mesaj_boyut then
       exit while
    end if
    gizlenecek = convert to decimal(mesaj_ikili(pos,bit+1:bit+4))
    if gizlenecek>=0 and gizlenecek<=3 then
      tasiyici_ikili(i,8)=0, tasiyici_ikili(i+1,8)=0
    else if gizlenecek>=4 and gizlenecek<=7 then
         tasiyici_ikili(i,8)=0, tasiyici_ikili(i+1,8)=1
    else if gizlenecek>=8 and gizlenecek<=11 then
         tasiyici_ikili(i,8)=1, tasiyici_ikili(i+1,8)=0
    else
         tasiyici_ikili(i,8)=1, tasiyici_ikili(i+1,8)=1
    end if
    if bit==4 then
      bit=0, pos=pos+1
    else
      bit=bit+4
    end if
    i=i+2
  end while
end procedure
```

Algoritma 2. YKKG Veri Çıkarma Yöntemi.

```
procedure VERICIKAR(Stego_Veri,geri_donme_esik)
  max_blok_sayi = sizeof(Stego_Veri)/2
  stego_ikili = convert to binary(Stego_Veri)
  geri_donen_deger[0]={0000,0100,1000,1100}
  geri_donen_deger[1]={0001,0101,1001,1101}
  geri_donen_deger[2]={0010,0110,1010,1110}
  geri_donen_deger[3]={0011,0111,1011,1111}
  mesaj_boyut = convert to decimal(LSB(20bit))
  i=1
  while i<max_blok_sayi do
    if sizeof(gizli_mesaj) >= mesaj_boyut then
       exit while
    end if
    gizli_veri_kodu=convert to decimal(stego_ikili(i,8), stego_ikili(i+1,8))
    mesaj_parcasi=geri_donen_deger[geri_donme_esik][gizleme_kodu]
    mesaj=mesaj+mesaj_parcasi
    if  uzunluk(mesaj)==8 then
       gizli_mesaj=gizli_mesaj+mesaj
       mesaj=""
    end if
    i=i+2
  end while
end procedure
```

3 Deneysel sonuçlar

İyi bir veri gizleme yöntemi, ilk olarak veri gizleme işlemi sonucunda taşıyıcı imgede gözle görülebilecek bozulmalara sebebiyet vermemesi ve imge ölçüt analizlerinden iyi sonuçlar üretmesi gerekmektedir. Bununla birlikte veri gizleme işleminin steganaliz yöntemlerince de algılanamaması yöntemin güvenliği için önemlidir. YKKG yönteminin imge kalitesini ölçmek için literatürde sıklıkla kullanılan tepe sinyal gürültü oranı (PSNR) [47] ve yapısal benzerlik kalite ölçütü (SSIM) [47],[48] kullanılmıştır. Örtü imgelerdeki bozulmaları ölçebilmek için Eş. 2'de verilen ortalama karesel hata (MSE) ve Eş. 3'de verilen tepe sinyal gürültü oranı (PSNR) kalite ölçütleri kullanılmıştır. Eş. 2'de m ve n imgenin satır ve sutun bilgilerini; O orjinal örtü imgeyi; S ise stego imgeyi temsil etmektedir. MSE değeri hesaplandıktan sonra bir sonraki adım PSNR hesaplamasıdır. PSNR örtü imge ile stego imge arasındaki benzerlik oranını ölçen bir imge ölçütüdür. Eş. 3'te MAX bir pikselin alabileceği maksimum değer ve genellikle 255'dir. Eş. 4'te SSIM kalite ölçütü verilmiştir. SSIM kalite ölçütünde yapısal olarak X ile Y vektörü arasında benzerlikler dikkate alınarak kalite indeksi hesaplanmaktadır. Eş. 4'te X orjinal taşıyıcı imgeyi, Y ise veri gizlenmiş imgeyi temsil etmektedir. Gizlenen verilerin boyutlarını gösterebilmek için Eş. 5'te verilen piksel başına bit (bit per pixel-bpp) ölçü birimi kullanılmıştır. Eş. 5'te piksel başına düşen bit sayısı imgedeki gizleme işleminde kullanılan bit sayısının imgedeki toplam bit sayısına oranıyla hesaplanmaktadır.

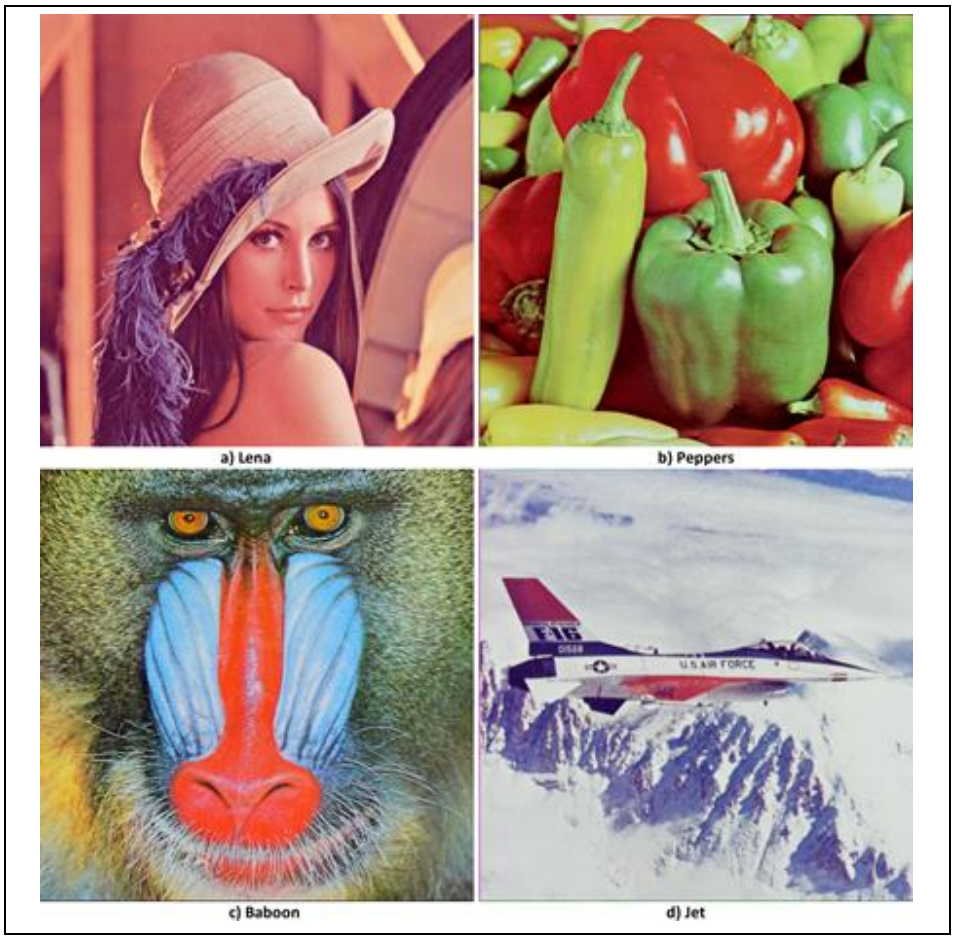
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[O(i,j) - S(i,j)]^2 \quad (2)$$

$$PSNR = 10log_{10}\left(\frac{MAX^2}{MSE}\right) \quad (3)$$

$$\text{SSIM (x, y)} = \frac{(2\,\mu x\,\mu y + C1)\,(2\,\sigma xy + C2)}{(\mu 2\,x + \mu 2\,y + C1)\,(\sigma\,2\,x + \sigma\,2\,y + C2)} \quad (4)$$

$$BPP = \frac{Gizlenen\ bit\ sayısı}{Toplam\ piksel\ sayısı} \quad (5)$$

Önerilen YKKG yönteminin steganaliz algoritmalarına karşı dayanıklılığını ölçebilmek için ki-kare ve komşuluk histogramı (neighbourhood histogram) yöntemleri kullanılmış ve sonuçları verilmiştir. Ayrıca önerilen yöntemin görsel ataklara karşı dayanıklılığını ölçebilmek için salt & pepper, gaussian, speckle ve poisson saldırı atakları kullanılmıştır. Geliştirilen YKKG yönteminin başarımını test etmek için Şekil 5'teki 512 x 512 x 24 boyutlarında 24-bit renkli literatürde sıklıkla kullanılan Lena, Pepper, Baboon ve Airplane taşıyıcı imgeleri kullanılmıştır.



Şekil 5. 512 x 512 x 24 Orijinal Test İmgeleri. (a): Lena. (b): Peppers. (c): Baboon. (d): Jet.

Figure 5. 512 x 512 x 24 Original Test Images. (a): Lena. (b): Peppers. (c): Baboon. (d): Jet.

Şekil 6(a)'da 512x512x24 boyutlarındaki orijinal peppers imgesine, Şekil 6(b)'deki 256x256x24 boyutlarındaki logo imgesi YKKG yöntemiyle gizlenmiştir. Şekil 6(c)'de veri gizlenmiş 512x512x24 boyutlarındaki stego imge ve Şekil 6(d), (e), (f) ve (g)'de farklı geri dönme eşik değerlerine göre geri çıkarılan 256x256x24 boyutlarındaki logo imgeleri verilmiştir. Şekil 6(d), (e), (f) ve (g)'deki geri çıkarılan imgeler incelendiğinde geri dönme eşik değeri arttıkça imgenin renk tonlamasındaki parlaklık artmaktadır. Geri dönme eşik değeri 1 seçiliyken geri çıkarılan Şekil 6(d)'deki imge koyu renk tonlarına sahipken, eşik değeri 2 seçiliyken Şekil 6(e)'deki imge daha açık, eşik değeri 3 seçiliyken Şekil 6(f)'deki imge daha da açık, eşik değeri 4 seçiliyken Şekil 6(g)'deki imge ise en açık rek tonuna sahiptir. Renk tonlarının bu şekilde değişmesinin sebebi, eşik değeri 4 seçildiğinde eşleştirme alanlarının en büyük değerleri geriye döndüğü için renk tonları parlak, eşik değeri 1 seçildiğinde ise eşleştirme alanlarının en küçük değerleri geri döndüğünden geriye çıkarılan imgenin renk tonları koyulaşmaktadır. Şekil 6(e)'de eşik değerin orta değer olan 2 değeri seçilmesinde, eşleştirme alanlarının orta değerleri geri döndüğü için orijinal gizlenen imgeye en yakın sonucu vermektedir. Geri dönme eşik değeri seçilirken, gizlenen imgenin renk tonuna göre kapalı tonlara sahip bir imge için 1 geri dönme eşiğinin, açık tonlara sahip bir imge içinse 4 geri dönme eşiğinin seçilmesi orijinal gizlenen imgeye en yakın oranda geri çıkarma sağlamaktadır.

Şekil 6'deki veri gizlenmiş Peppers ve farklı eşik değerleri ile geri çıkarılan imgeleri orijinal imgelerle kıyaslamak için PSNR ve SSIM kalite ölçütleri kullanılmış ve Tablo 2'de sonuçlar verilmiştir.

Şekil 6. YKKG yöntemi ile veri gizleme ve çıkarma. (a): Orijinal peppers 512x512x24. (b): Gizlenen logo imgesi 256x256x24. (c): Veri gizlenmiş peppers 512 x 512 x 24. (d): Çıkarılan imge 256 x 256 x 24, eşik değeri=1. (e): Çıkarılan imge, eşik değeri=2. (f): Çıkarılan imge, eşik değeri=3. (g): Çıkarılan imge, eşik değeri=4.

Figure 6. Data hiding and extraction using YKKG method. (a): Original peppers 512x512x24. (b): Hidden logo image 256x256x24. (c): data hidden peppers 512 x 512 x 24. (d): Extracted image 256 x 256 x 24, threshold = 1. (e): Extracted image, threshold = 2. (f): Extracted image, threshold = 3. (g): Extracted image, threshold = 4.

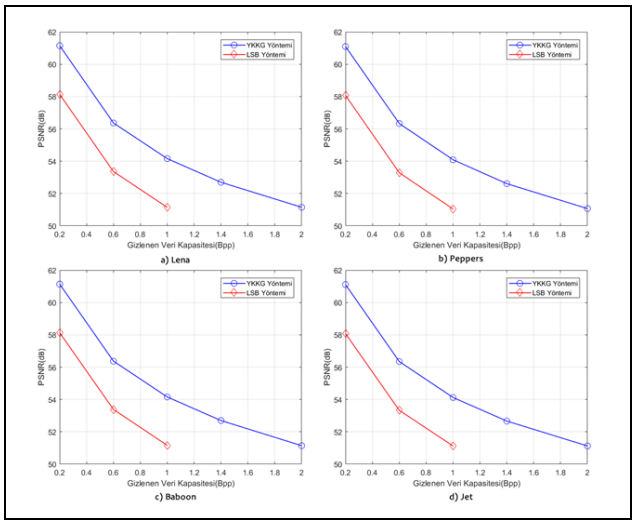
Tablo 2. Orijinal ve veri gizlenmiş Peppers imgeleri ile geri dönme eşiğine göre geri çıkarılan imgelerin PSNR ve SSIM değerleri.

Table 2. PSNR and SSIM values of original and data hidden Peppers images and images extracted according to the recovery threshold.

	Orijinal ve Veri Gizlenmiş Peppers	Geri Dönme Eşiği			
		1	2	3	4
PSNR	51,067	15,38	19,00	22,95	21,90
SSIM	0,9961	0,889	0,9382	0,9481	0,9388

PNSR kalite ölçütü orjinal imge ile veri gizlenmiş imgenin arasındaki benzerliğe bakarak gizleme sonucunda oluşan gürültüyü desibel (dB) biriminde hesaplar [47]. Bir imge için 40 dB ve üstü iyi bir başarı değeridir. Şekil 6(a) ve (c)'deki orijinal ve veri gizlenmiş Peppers imgelerinde 51,067dB PSNR ve 0,996 SSIM değeri elde edilmiştir. PSNR ve SSIM değerlerine göre YKKG veri gizleme yöntemi, 512x512x24 boyutlarındaki orijinal Peppers imgesinin yarısı olan 256x256x24 boyutlarındaki logo imgesini, taşıyıcı imgede düşük bozulumla ve geleneksel LSB yöntemine göre iki kat veri gizleme oranı ile başarılı bir şekilde gizlemiştir. Şekil 6(d)'de eşik değeri 1 seçilerek geri çıkarılan logo imgesi ile orijinal logo imgesi arasında 15,38 dB PSNR, 0,8890 SSIM değeri, eşik değeri 2 seçilerek geri geri çıkarılan logo imgesi 19,00 dB PSNR, 0,9382 SSIM değeri, eşik değeri 3 seçilerek geri geri çıkarılan logo imgesi 22,95 dB PSNR, 0,9481 SSIM değeri, eşik değeri 4 seçilerek geri çıkarılan logo imgesi 21,90 dB PSNR ve 0,9388 SSIM değeri elde edilmiştir. Bu sonuçlara göre de YKKG yöntemiyle eşik değeri 3 seçilerek geri çıkarma işlemi yapıldığında daha başarılı sonuç vermektedir.

Orijinal test imgelerine hem YKKG yöntemiyle hemde LSB yöntemiyle gizlenen veri kapasitesine göre PSNR değerlerini gösteren grafikler Şekil 7'da verilmiştir. Testlerde tüm imgelere yöntemin gizleyebileceği maksimum veri gizleme oranlarında, LSB yönteminde 0-1bpp, YKKG yönteminde ise 0-2 bpp oranında veri gizlenmiştir.



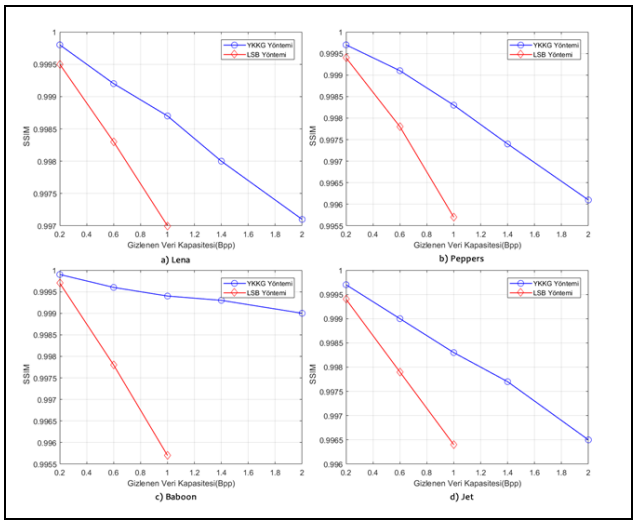
Şekil 7. Gizlenen Veri Kapasitesine göre PSNR değerleri. (a): Lena. (b): Peppers. (c): Baboon. (d): Jet.

Figure 7. PSNR values according to Hidden Data Capacity.

(a): Lena. (b): Peppers. (c): Baboon. (d): Jet.

Şekil 7(a) Lena imgesinde, 1 bpp oranında veri gizlendiğinde LSB yöntemiyle 51.5dB PSNR değeri elde edilirk en, YKKG yönteminde 54.1dB PSNR değeri elde edilmektedir. Peppers, Baboon ve Jet imgelerinde de PSNR değerleri aynı çıkmaktadır. Bu sonuçlara göre YKKG yöntemi geleneksel LSB yöntemine göre hem 2 kat veri gizlemekte hem de imgede daha başarılı PSNR değeri elde edilmektedir.

Orijinal test imgelerine hem YKKG yöntemiyle hemde LSB yöntemiyle gizlenen veri kapasitesine göre SSIM değerlerini gösteren grafikler Şekil 8'de verilmiştir. Şekil 7'da ki PSNR değerleri ile paralel olarak SSIM değerleri de aynı sonuçları vermektedir. Özellikle Şekil 8(c)'de Baboon imgesinde 2bpp oranında veri gizlenmesine rağmen 1bpp oranında veri gizleyen LSB yöntemine göre daha iyi bir SSIM değeri elde edilmiştir. 1bpp oranında veri gizleyen LSB yönteminde 0,9956 SSIM değeri elde edilirken, 2bpp oranında veri gizleyen YKKG yönteminde 0,9990 SSIM değeri elde edilmiştir. SSIM değerlerine görede YKKG yöntemi 2 kat oranında veri gizlemesine rağmen imgede daha az bozulma oluşturmaktadır.



Şekil 8. Gizlenen veri kapasitesine göre SSIM değerleri. (a): Lena. (b): Peppers. (c): Baboon. (d): Jet.

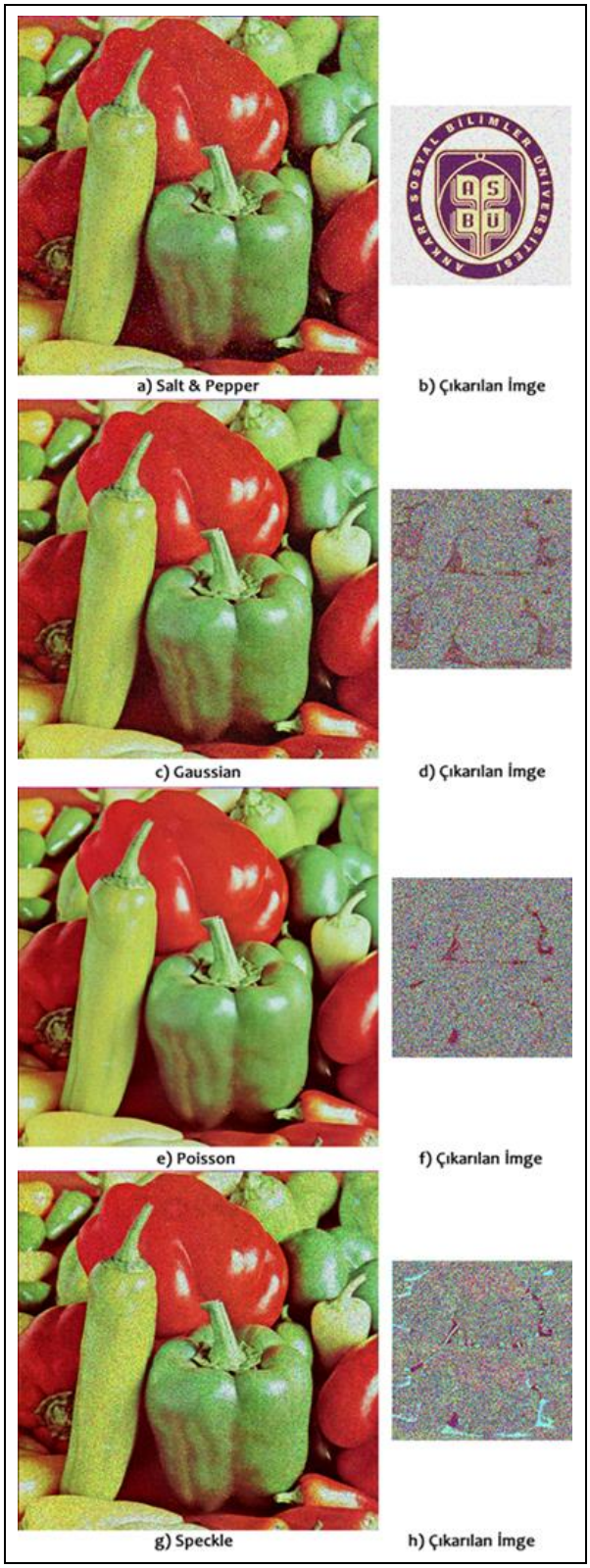
Figure 8. SSIM values according to Hidden Data Capacity.

(a): Lena. (b): Peppers. (c): Baboon. (d): Jet.

Şekil 9'da YKKG yönteminin ataklara karşı başarımını gösteren test sonuçları verilmiştir. 256 x 256 x 24 boyutlarındaki Logo imgesinin YKKG yöntemiyle gizlendiği 512 x 512 x 24 boyutlarındaki Peppers imgesine Salt & Peppers, Gaussian, Poisson ve Speckle saldırıları yapılmıştır. Her saldırı sonrasında Peppers imgesi ve geri çıkarılan Logo imgesi verilmiştir. Salt & Peppers saldırısında Şekil 9(a) ve (b) incelendiğinde Peppers imgesi görünür şekilde bozulmaya uğramasına rağmen içinde gizlenen imgenin bilgileri saldırıdan etkilenmiş, ancak anlaşılmaz şekilde bozulmamıştır. Diğer tüm saldırılarda Peppers imgesinde görünürde aşırı derecede bozulma oluşmazken aslında LSB bitlerinde tahribat büyüktür. Bu nedenle geri çıkarılan Logo imgesinde yüksek oranda bozulma oluşmuştur.

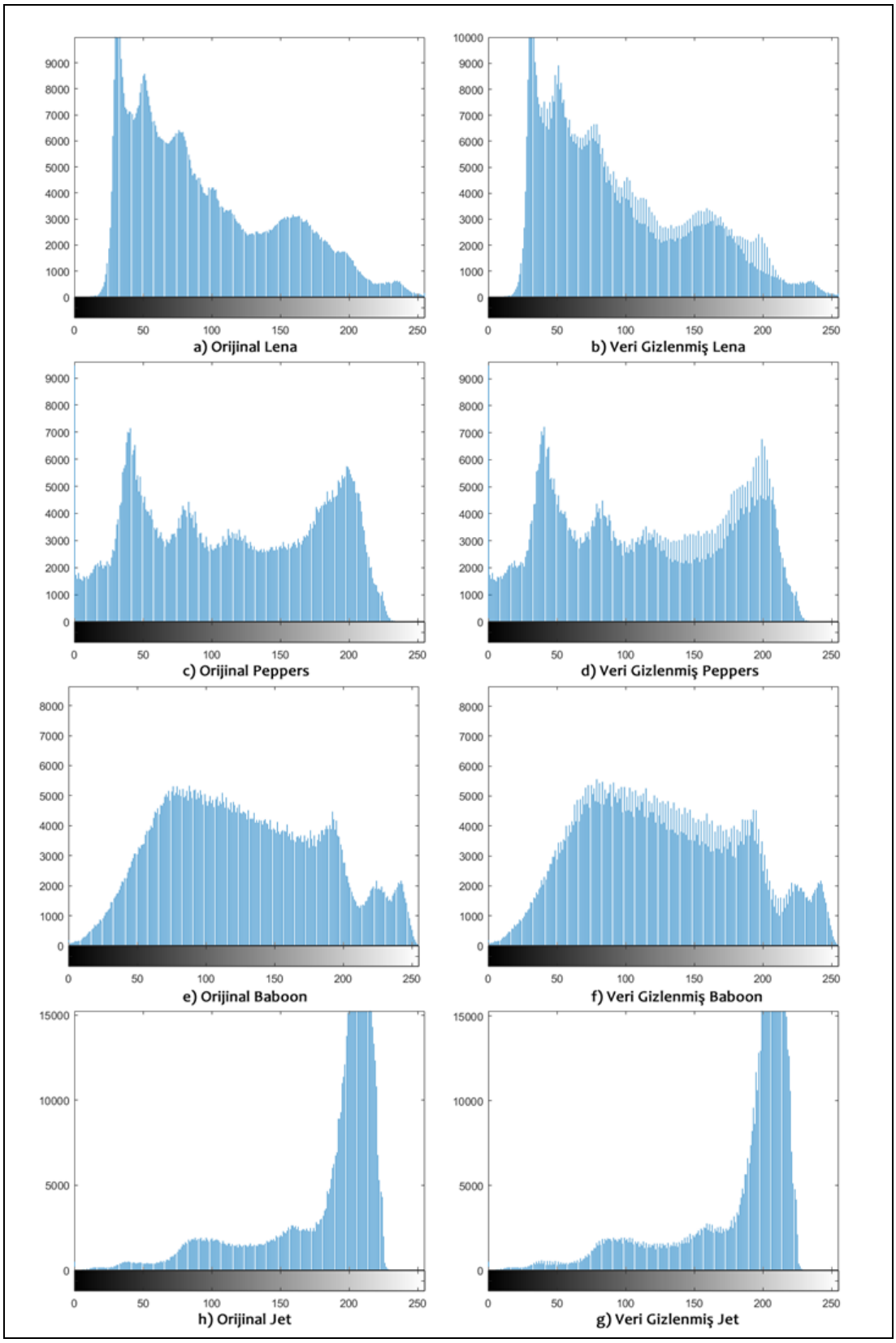
Salt & Peppers saldırısının yoğunluğu diğer saldırılara göre daha azdır. Bu da LSB bitlerini daha az tahrip etmiştir. Diğer saldırılar daha yoğun bir şekilde imgenin LSB bitlerinde tahribat oluşturmaktadır. Şekil 9(d), (f) ve (g)'de de görüldüğü gibi çıkarılan Logo imgeleri saldırılardan yüksek oranda etkilenmiştir.

Orijinal test imgelerine 19,6 KB veri gizlenmiş ve imgelerinin histogram grafikleri Şekil 10'da verilmiştir. Şekil 10'daki histogramlar incelendiğinde veri gizlenmiş imgelerin histogramlarında fark gözlenmemektedir. Histogram grafiklerine göre de YKKG yöntemi imgede düşük oranda bozulma oluşturmaktadır.



Şekil 9. Şekil 9. YKKG yöntemiyle 256x256x24 Logo imgesi gizlenmiş Peppers imgelerine yapılan ataklar ve ataklı imgelerden çıkarılan imgeler. (a): Salt & Pepper. (b): Çıkarılan imge. (c): Gaussian. (d): Çıkarılan imge. (e): Poisson. (f): Çıkarılan imge. (g): Speckle. (h): Çıkarılan imge.

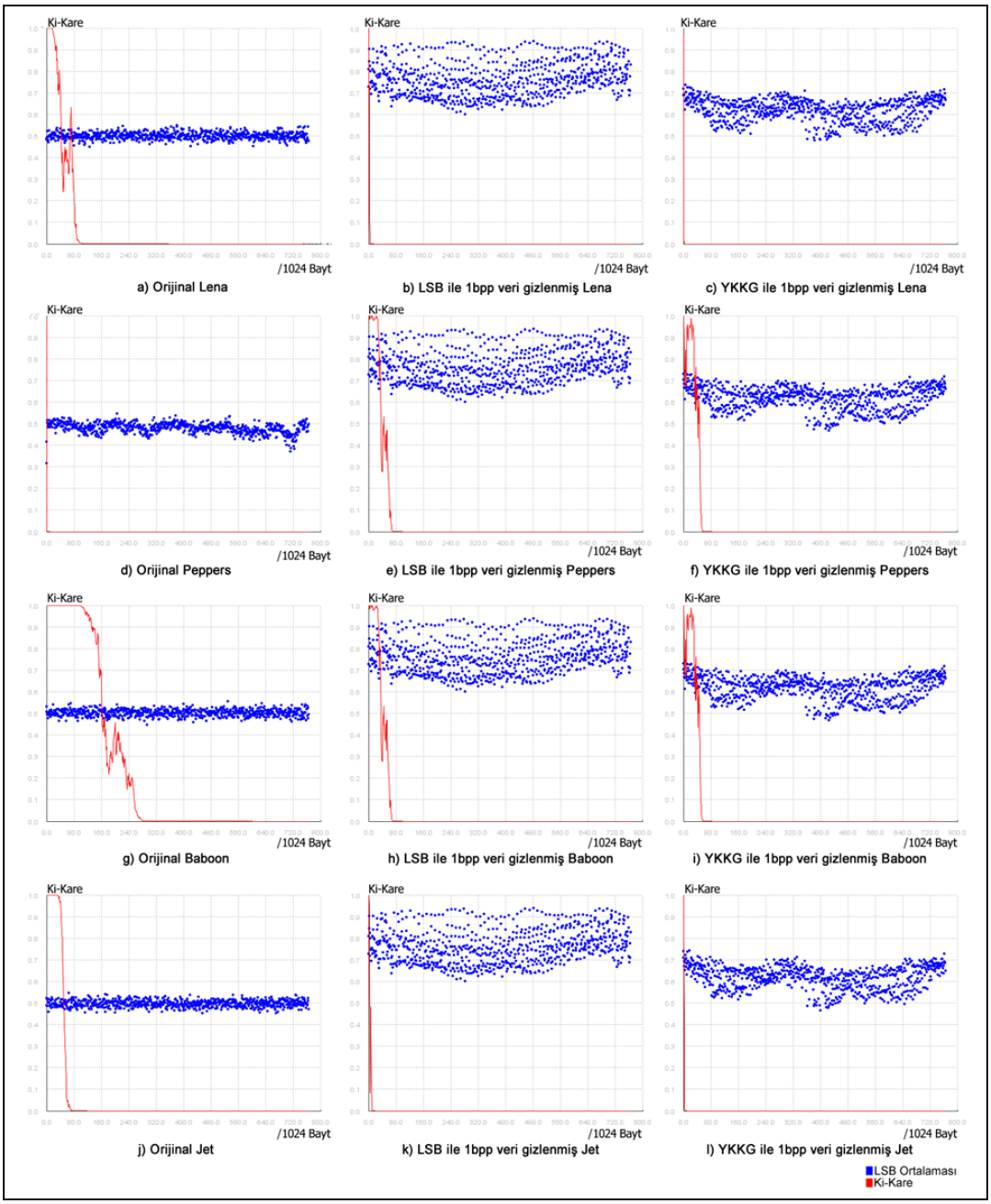
Figure 9. Attacks made to Peppers images with the 256x256x24 logo image hidden by YKKG method and extracted images from attacked images.(a): Salt & Pepper. (b): Extracted image. (c): Gaussian. (d): Extracted image. (e): Poisson. (f): Extracted image. (g): Speckle. (h): Extracted image.



Şekil 10. Orijinal ve YKKG yöntemiyle 19,6 KB veri gizlenmiş test imgelerinin histogramları. (a): Orijinal lena. (b): Veri gizlenmiş lena. (c): Orijinal peppers. (d): Veri gizlenmiş peppers. (e): Orijinal baboon. (f): Veri gizlenmiş baboon. (h): Orijinal jet. (g): Veri gizlenmiş Jet.

Figure 10. Histograms of test images with 19,6 KB data hidden using original and YKKG method (a): Original lena. (b): Data hidden lena. (c): Original peppers. (d): Data hidden peppers. (e): Original baboon. (f): Data hidden baboon. (h): Original jet. (g): Data hidden jet.

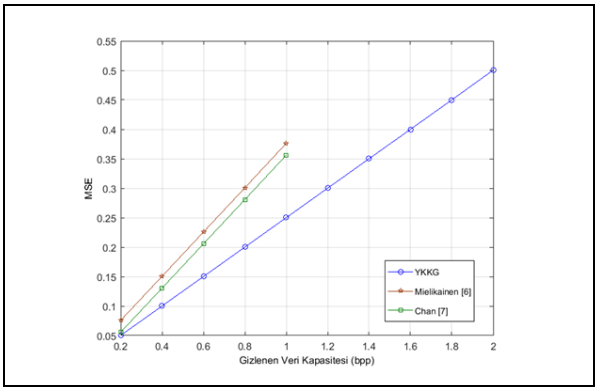
Orijinal test imgelerine YKKG ve LSB yer değiştirme yöntemleriyle 1bpp veri gizlenmesiyle oluşan stego imgelerin ki-kare ve LSB dağılımlarını gösteren grafikler Şekil 11'da verilmiştir. Şekil 11'daki grafikler incelendiğinde orjinal test imgelerinin ve YKKG ile 1bpp veri gizlenmiş imgelerin ki-kare sonuçları birbirine yakın çıktığı gözlenmektedir. Ayrıca tüm grafiklerde LSB ortalama değerlerinin dağılımları incelendiğinde de YKKG ile 1bpp veri gizlenmiş imgelerin LSB dağılımları orjinal imgelerin LSB dağılımlarına daha yakın çıkarken, LSB yer değiştirme yöntemi ile 1bpp veri gizlenmiş imgelerin LSB dağılımları orjinal imgelerin LSB dağılımlarına göre dağınık olarak yayıldığı gözlenmiştir.



Şekil 11. Orijinal test imgelerine YKKG ve LSB yöntemleriyle 1bpp veri gizlenmesiyle oluşan stego imgelerinin ki-kare ve LSB dağılım grafikleri a) Orijinal lena b) LSB ile 1bpp veri gizlenmiş lena c) YKKG ile 1bpp veri gizlenmiş lena d) Orijinal peppers e) LSB ile 1bpp veri gizlenmiş peppers f) YKKG ile 1bpp veri gizlenmiş peppers g) Orijinal baboon h) LSB ile 1bpp veri gizlenmiş baboon i) YKKG ile 1bpp veri gizlenmiş baboon j) Orijinal jet k) LSB ile 1bpp veri gizlenmiş jet l) YKKG ile 1bpp veri gizlenmiş jet.

Figure 11. Chi-square and LSB scatter plots of stego images created by 1bpp data hiding with YKKG and LSB methods on original test images a) Original Lena b) 1bpp data hidden with LSB lena c) 1bpp data hidden with YKKG lena d) Original peppers e) 1bpp data with LSB peppers hidden f) 1bpp data hidden with YKKG peppers g) Original baboon h) 1bpp data hidden with LSB baboon i) 1bpp data hidden with YKKG baboon j) Original jet k) 1bpp data hidden with LSB jet l) 1bpp data hidden jet with YKKG.

Önerilen YKKG yönteminin veri gizleme başarımını literatürdeki eşdeğer iki çalışma ile kıyaslayabilmek için 150 adet 500x400 boyutlarında 24-bit renkli imgelere YKKG yöntemi ve literatürdeki eşdeğer iki çalışma ile 15KB (0,2bpp) ve katları olacak şekilde 10 kez maksimum 150KB (2bpp) veri gizlenmiştir. İmgeler Internet üzerinden indirilmiş uçak, hayvan, çiçek ve balık görselleri içermektedir. Kullanılan imgelere ve geliştirilen yöntemin Matlab® kodlarına Internet üzerinden erişilebilir [49]. Her gizleme yönteminde imgelerdeki bozulmalar Eş. 2 ve 3'de verilen MSE ve PSNR ölçüm parametreleri ile ölçülmüştür. Literatürdeki Mielikainen [6] ve Chan [7] yöntemi dosyaya en fazla LSB bit gizleme sınırı kadar yani dosya boyutunun 1/8'i kadar oranda veri gizleyebildiği için maksimum 75KB (1bpp) veri gizleyebilmektedir. Ancak, YKKG yöntemi dosya boyutunun ¼'ü (2bpp) kadar oranda veri gizleyebilmektedir. Önerilen YKKG yöntemi, Mielikainen [6] ve Chan [7] yöntemleri ile 150 imgeye gizlenen veriler MSE kalite ölçütü ile analiz edilerek hesaplanan ortalama değerler Şekil 12'de verilmiştir.

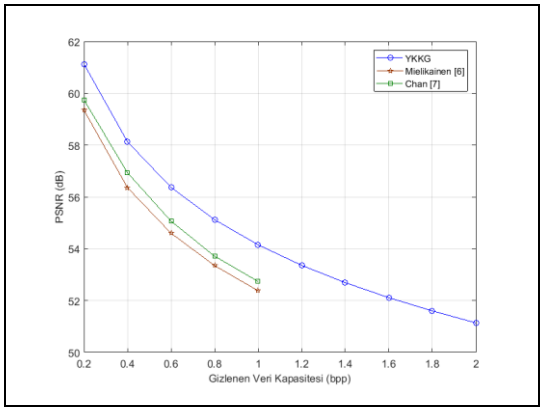


Şekil 12. 150 imgeye önerilen yöntem ve literatür çalışmalarıyla farklı oranlarda veri gizlemeleriyle oluşan stego imgelerin MSE değerleri ortalamalarının karşılaştırılması.

Figure 12. Comparison of the average MSE values of the stego images formed with different rates of data hiding with the proposed method and literature studies on 150 images

Şekil 12'deki grafiğe göre en başarılı yöntemin YKKG yöntemi olduğu görülmektedir. İmgelere 1bpp oranında veri gizlendiği durumda Mielikainen [6] 0,375 MSE değeri elde ederken Chan [7] 0,36 ve YKKG yöntemi ise 0,25 ile en düşük MSE değeri elde etmiştir. YKKG yöntemi 1,4bpp veri gizlediği durum ile Chan [7]'in 1bpp veri gizlediği durumda aynı oranda MSE değeri elde etmişlerdir. YKKG yöntemi daha fazla veri gizlerken bile Chan [7]'in değerini yakalamıştır. Sonuçlar YKKG yönteminin imgede daha az değişiklik yaptığını kanıtlamaktadır. YKKG yöntemi 2bpp oranında veri gizleyebilirken diğer yöntemlerin 1bpp sınırında kaldıkları gözlenmektedir. Buda YKKG yönteminin kapasitesinin literatürdeki yöntemlerin iki katı olduğunu göstermektedir.

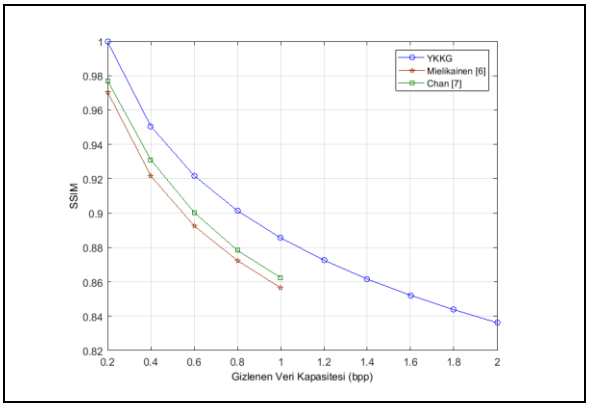
Önerilen YKKG yöntemi, Mielikainen [6] ve Chan [7] yöntemleri ile 150 imgeye gizlenen veriler PSNR kalite ölçütü ile analiz edilerek hesaplanan ortalama değerler Şekil 13'te verilmiştir.Veri gizlenmiş bir imgenin ölçülen PSNR değeri yükseldikçe orjinal örtü imge ile arasındaki benzerlikte o oranda yüksek demektir. Başka bir ifadeyle PSNR değerinin yüksek çıkması, gizleme işlemi yapan steganografi yönteminin örtü imgede daha az değişiklik yaptığını göstermektedir. 1bpp veri gizlendiği durumda YKKG yöntemi 55 dB, Chan [7] 52,75 dB ve Mielikainen [6] 52,5 dB PSNR değeri elde etmiştir. Şekil 13'teki grafiğe göre PSNR değeri en yüksek çıkan YKKG yönteminin örtü imgede en az değişiklik yapma konusunda en başarılı sonucu veren yöntem olduğu görülmektedir.



Şekil 13. 150 imgeye önerilen yöntem ve literatür çalışmalarıyla farklı oranlarda veri gizlemeleriyle oluşan stego imgelerin PSNR değerleri ortalamalarının karşılaştırılması.

Figure 13. Comparison of the average PSNR values of the stego images formed with different rates of data hiding with the proposed method and literature studies on 150 images

Önerilen YKKG yöntemi, Mielikainen [6] ve Chan [7] yöntemleri ile 150 imgeye gizlenen veriler SSIM kalite ölçütü ile analiz edilerek hesaplanan ortalama değerler Şekil 14'te verilmiştir.



Şekil 14. 150 imgeye önerilen yöntem ve literatür çalışmalarıyla farklı oranlarda veri gizlemeleriyle oluşan stego imgelerin SSIM değerleri ortalamalarının karşılaştırılması.

Figure 14. Comparison of the average SSIM values of the stego images formed with different rates of data hiding with the proposed method and literature studies on 150 images

1 bpp veri gizlendiği durumda YKKG yöntemi 0,89, Chan [7] 0,865 ve Mielikainen [6] 0,858 PSNR değeri elde etmiştir. Şekil 14'teki grafiğe göre SSIM değeri en yüksek çıkan YKKG yönteminin örtü imgede en az değişiklik yapma konusunda en başarılı sonucu veren yöntem olduğu görülmektedir.

Sonuç olarak önerilen YKKG yöntemi tüm bu deneysel sonuçlar ışığında örtü imgede en az değişiklik yaparak yüksek kapasitede veri gizlemeyi başarmaktadır. Önerilen YKKG yöntemi literatürdeki Mielikainen [6] ve Chan [7] yöntemlerine göre iki kat kapasitede veri gizlerken daha yüksek PSNR değeri ile örtü imgede daha az bozulma oluşturmaktadır. Yöntemin ki-

kare, histogram gibi bilinen steganaliz ataklarına karşı başarısıda ispatlanmıştır.

Önerilen yöntemin avantajları ve dezavantajları bulunmaktadır.

Avantajlar;

1- Önerilen yöntem, 24-bit imgeye 24 bit veri gizlerken 24-bit taşıyıcı imgeyi 4-bit parçalara bölmekte ve her bir parçayı 2-bit gizleme kodu ile gizlediği için yarı yarıya düşük boyutta veri gizlemekte ve buda kapasite olarak iki kat verim sağlamaktadır. Böylece literatürdeki benzer çalışmalar ile örtü imgeye aynı oranda veri gizlediğinde yarı yarıya daha az veri gizlemektedir,
2- Önerilen veri gizleme yöntemi LSB yöntemi kullanan literatürdeki çalışmalara göre aynı oranda veri gizlendiğinde taşıyı ortama daha az oranda veri gizlediği için örtü imgede daha düşük bozulma oluşturmaktadır. Bu nedenle daha yüksek PSNR ve SSIM değerleri elde etmektedir,
3- Önerilen veri gizleme yöntemi LSB yöntemi kullanan literatürdeki benzer çalışmalara kıyasla iki kat daha fazla veri kapasitesi sunmaktadır,
4- Önerilen yöntem daha az veri gizlediği içinrtü imgedeki düşük bozulum oluşumu nedeniyle Ki-kare ve histogram steganaliz yöntemlerine karşı ö algınamazlık oranı yüksek olmakta ve sonuçlarda daha başarılı çıkmaktadır.

Dezavantajları ise;

1- Salt & pepper, gaussian, speckle ve poisson saldırıları, bit uzayı kullanan yöntemleri etkilediği için önerilen yöntemin gizlediği gizli veriyide etkilemektedir. Gizli veri geri çıkarıldığında bozulmalar oluşmaktadır. Bu önerilen yöntemin zayıflığından değil, kullanılan bit uzayı yönteminin zayıflığından kaynaklanmaktadır. Literatürde bit uzayı yöntemi kullanan tüm yöntemlerde ortak sorun bulunmaktadır,
2- Önerilen yöntem örtü imgede daha az değişiklik yapmak için gizli imgeyi kayıplı olarak gizlemektedir. Yöntem veri gizleme alanını kısıtlı olduğu durumlarda yüksek miktarda veri gizlemek için kullanılabilir.

4 Sonuçlar

Bu çalışmada, 24-bit renkli imge içine 24-bit renkli imgeyi kayıplı gizleyen yüksek kapasiteli, düşük bozulumlu ve tersinir yeni bir veri gizleme yöntemi (YKKG) önerilmiştir. Önerilen yöntem geleneksel LSB yöntemlerine ve literatürdeki çalışmalara göre iki kat veri gizlerken örtü imgede daha düşük bozulma oluşturmaktadır. Yapılan testlere göre önerilen yöntem, yüksek PSNR ve SSIM değerleri elde etmektedir. Bu sebeble yüksek algılanamazlık, kapasite ve güvenlik sağlamaktadır. Ayrıca önerilen veri gizleme yöntemi tersinir olduğundan geri çıkarma işleminde orijinal imgeye ihtiyaç duymamaktadır.

Elde edilen sonuçlar aşağıda sıralanmıştır:

- Önerilen yöntem, 24-bit imgeyi gizlerken 4-bit parçalara bölmekte ve her bir parçayı 2-bit gizleme kodu ile gizlediği için yarı yarıya düşük boyutta veri gizlemekte ve buda kapasite olarak iki kat verim sağlamaktadır,
- Önerilen yöntem ve literatür çalışmalarıyla 150 imgeye farklı oranlarda veri gizlenmesiyle oluşan stego imgelerin test sonuçlarına göre önerilen yöntem LSB yöntemi kullanan literatürdeki çalışmalara göre iki kat veri gizleyebilmektedir,
- Yöntem aynı oranda veri gizlendiği durumda LSB yöntemi kullanan literatürdeki çalışmalara göre daha az oranda veri gizlediği için örtü imgede daha düşük bozulma oluşturmaktadır. Bu nedenle daha yüksek PSNR ve SSIM değerleri elde etmektedir,
- Ki-kare ve histogram steganaliz yöntemlerine karşı örtü imgedeki düşük bozulum oluşumu nedeniyle algınamazlık oranı yüksek olmakta ve sonuçlar başarılı çıkmaktadır,
- Salt & pepper, gaussian, speckle ve poisson saldırıları, önerilen yöntemin veri çıkarma başarımını LSB yöntemlerini kullanan tüm yöntemlerin etkilediği kadar etkilemiş, bu açıdan sonuç literatürdeki çalışmalar ile aynı olmaktadır,
- Önerilen yöntem örtü imgede daha az değişiklik yapmak için gizli imgeyi kayıplı olarak gizlemektedir. Yöntem veri gizleme alanını kısıtlı olduğu durumlarda yüksek miktarda veri gizlemek için kullanılabilir.

Gelecekte kayıplı gizleme nedeniyle oluşan geri çıkarılan imgedeki kalite bozulumunu en aza indirgemek için çalışma yapılacaktır.

5 Conclusions

In this study, a new high-capacity, low-distortion and reversible data hiding method (YKKG) is proposed that hides a 24-bit color image into a 24-bit color image with loss. While the proposed method hides twice the data compared to traditional LSB methods and studies in the literature, the cover creates less distortion in the image. According to the tests performed, the proposed method achieves high PSNR and SSIM values. For this reason, it provides high imperceptibility, capacity and security. In addition, since the proposed data hiding method is reversible, it does not need the original image for extract.

The results obtained are listed below:

- The proposed method, while hiding the 24-bit image, divides it into 4-bit segments and hides half the data with a 2-bit hiding code, thus providing twice the efficiency in capacity,
- According to the test results of stego images formed by hiding data at different rates to 150 images with the proposed method and literature studies, the proposed method can hide data twice as compared to the studies in the literature using LSB method,
- When the method hides data at the same rate, the cover creates less distortion in the image because it hides less data than the studies in the literature using the LSB method. Therefore, it achieves higher PSNR and SSIM values,
- Against chi-square and histogram steganalysis methods, the rate of non-perception is high due to low distortion in the cover image and the results are successful,
- Salt & pepper, gaussian, speckle and poisson attacks affected the data extraction performance of the proposed method as much as all methods using LSB methods, so the result is the same as the studies in the literature,
- The proposed method lossy hides the hidden image in order to make less change in the cover image. The method can be used to hide large amounts of data in cases where the data hiding space is limited.

In the future, efforts will be made to minimize the quality deterioration in the retrieved image due to lossy concealment.

6 Yazar katkı beyanı

Gerçekleştirilen çalışmada Ali Durdu fikrin oluşması, tasarımın yapılması, literatür taraması, elde edilen sonuçların değerlendirilmesi, kullanılan malzemelerin temin edilmesi ve sonuçların incelenmesi, yazım denetimi ve içerik açısından makalenin kontrol edilmesi başlıklarında katkı sunmuştur.

7 Etik kurul onayı ve çıkar çatışması beyanı

Hazırlanan makalede etik kurul izni alınmasına gerek yoktur.

Hazırlanan makalede herhangi bir kişi/kurum ile çıkar çatışması bulunmamaktadır.

8 Kaynaklar

[1] Durdu A. İmge İçerisine LSB Eşleştirme Alanı Tabanlı Kayıplı İmge Gizleyen Yüksek Kapasiteli Tersinir Sırörtme Yöntemi. Doktora Tezi, Sakarya Üniversitesi, Sakarya, Türkiye, 2016.

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

[3] Durdu A. Sırörtülü Ses Dosyalarının Yapay Zekâ Yöntemleri Yardımıyla Çözümlenmesi, Yüksek Lisans Tezi, Sakarya Üniversitesi, Sakarya, Türkiye, 2007.

[4] Yalman Y. Sayısal Görüntüler İçin Histogram Temelli Veri Gizleme Yöntemi Ve Uygulama Yazılımı. Doktora Tezi, Kocaeli Üniversitesi, Kocaeli, Türkiye, 2010.

[5] Vural C, Baraklı B. "Reversible video watermarking using motion-compensated frame interpolation error expansion". *Signal, Image and Video Processing*, 9, 1613-1623, 2015.

[6] Sharp T. "An Implementation of Key-Based Digital Signal Steganography". Information Hiding. *Lecture Notes in Computer Science*, Springer, Berlin, Germany, 24 October 2001.

[7] Mielikainen J. "LSB matching revisited". *IEEE Signal Processing Letters*, 13(5), 285-287, 2006.

[8] Chan C. "On using LSB matching function for data hiding in pixels". *Fundamenta Informaticae*, 96(1-2), 49-59, 2009.

[9] Tian J. "Reversible data embedding using a difference expansion". *IEEE Transactions on Circuits and Systems*, 13(8), 890-896, 2003.

[10] Alattar AM. "Reversible watermark using the difference expansion of a generalized integer transform". *IEEE Transactions on Image Processing*, 13(8), 1147-1156, 2004.

[11] Chang CC, Chou Y, Kieu TD. "Information hiding in dual images with reversibility". *Third International Conference on Multimedia and Ubiquitous Engineering,* Qingdao, China, 4-6 June 2009.

[12] Lu T, Tseng C, Wu J. "Dual imaging-based reversible hiding technique using LSB matching". *Signal Processing*, 108, 77-89, 2015.

[13] Ker AD. "Quantitive evaluation of pairs and RS steganalysis". *Security, Steganography, and Watermarking of Multimedia Contents VI, California, United States,* 22 June 2004.

[14] Wu D, Tsai W. "A steganographic method for images by pixel-value differencing". *Pattern Recognition Letters,* 24(9-10), 1613-1626, 2003.

[15] Wang CM, Wu NI, Tsai CY, Hwang MS. "A high quality steganographic method with pixel-value differencing and modulus function". *Journal of Systems and Software*, 81(1), 150-158, 2008.

[16] Fridrich J, Soukal D. "Matrix Embedding for Large Payloads". *IEEE Transactions on Information Forensics and Security*, 1(3), 390-395, 2006.

[17] Kurtuldu O, Arica N. "A new steganography method using image layers". *23rd International Symposium on Computer and Information Sciences*, Istanbul, Turkey, 27-29 October 2008.

[18] Wu H, Wang H, Hu Y, Zhou L. "Efficient reversible data hiding based on prefix matching and directed LSB embedding". *Digital-Forensics and Watermarking: 13th International Workshop*, Taipei, Taiwan, 1-4 October, 2014.

[19] Wikipedia. "Simetrik Anahtar Algoritmalar". https://tr.wikipedia.org/wiki/Simetrik_anahtar_algoritmalar (30.12.2020).

[20] Wikipedia. "Diffie-Hellman Anahtar Değişimi". https://tr.wikipedia.org/wiki/Diffie-Hellman_anahtar_değişimi (30.12.2020).

[21] Huang F, Zhong Y, Huang J. "Improved algorithm of edge adaptive image steganography based on LSB matching revisited algorithm". *Lecture Notes in Computer Science*, 8389, 19-31, 2014.

[22] Sabeti V, Samavi S, Shirani S. "An adaptive LSB matching steganography based on octonary complexity measure". *Multimedia Tools and Applications*, 64(3), 777-793, 2013.

[23] Jain R, Kumar N. "Efficient data hiding scheme using lossless data compression and image steganography". *International Journal of Engineering Science and Technology,* 4(8), 3908-3915, 2012.

[24] Atıcı MA, Sağıroğlu Ş. "Development of a new folder lock appraoch and software based on steganography". *Journal of the Faculty of Engineering and Architecture of Gazi University*, 31(1), 129-144, 2016.

[25] Emek S. Sabit Görüntüler ve Video Hareketleri İçin Ayrık Dalgacık Dönüşümlü Ayrık Kosinus Dönüşümlü Tabanlı Sayısal Damgalama Yöntemi. Doktora Tezi, Yıldız Teknik Üniversitesi, İstanbul, Türkiye, 2006.

[26] Boland FM, O'Ruanaidh JJK, Dautzenberg, C. "Watermarking digital images for copyright protection". *Fifth International Conference on Image Processing and its Applications*, Edinburgh, Scotland, 4-6 July 1995.

[27] Podilchuk CI, Zeng W. "Digital image watermarking using visual models". *Human Vision and Electronic Imaging II,* California, United States, 3 June 1997.

[28] Hernández JR, Amado M, Pérez-González F. "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure". *IEEE Transactions on Image Processing*, 9(1), 55-68, 2000.

[29] Hartung F, Girod B. "Watermarking of uncompressed and compressed video". *Signal Processing*, 66(3), 283-301, 1998.

[30] Bhatnagar G, Raman B. "A new robust reference watermarking scheme based on DWT-SVD". *Computer Standards and Interfaces*, 31(5), 1002-1013, 2009.

[31] Zeng W, Hu R, Ai H. "Audio steganalysis of spread spectrum information hiding based on statistical moment and distance metric", *Multimedia Tools and Applications*, 55(3), 525-556, 2011.

[32] Baby D, Thomas J, Augustine G, George E, Michael NR. "A novel DWT based ımage securing method using steganography". *Procedia Computer Science*, 46, 612-618, 2015.

[33] Chen P, Lin H, "A dwt based approach for image steganography". *International Journal of Applied Science and Engineering*, 4(3), 275-290, 2006.

[34] Kamila S, Roy R, Changder S. "A DWT based steganography scheme with image block partitioning". *2nd International Conference on Signal Processing and Integrated Networks*, New Delphi, India, 19-20 February 2015.

[35] Bhattacharyya D, Kim T. "Image Data Hiding Technique Using Discrete Fourier Transformation". *International Conference on Ubiquitous Computing and Multimedia Applications*, Daejeon, Korea, 13-15 April 2011.

[36] Chen W. "Color image steganography scheme using DFT, SPIHT codec, and modified differential phase-shift keying techniques". *Applied Mathematics and Computation*, 196(1), 40-54, 2008.

[37] Tuncer T, Avcı E. "Göktürk alfabesi tabanlı görsel sır paylaşımı metodu ile veri gizleme uygulaması". *Journal of the Faculty of Engineering and Architecture of Gazi University*, 31(3), 781-789, 2016.

[38] Chen K, Zhou H, Zhou W, Zhang W, Yu N. "Defining cost functions for adaptive JPEG steganography at the microscale". *IEEE Transactions on Information Forensics and Security*, 14(4), 1052-1066, 2019.

[39] Sarmah DK, Kulkarni AJ. "Improved cohort ıntelligence-a high capacity, swift and secure approach on JPEG image steganography". *Journal of Information Security and Applications*, 45, 90-106, 2019.

[40] Khosravi B, Khosravi B, Khosravi B, Nazarkardeh K. "A new method for pdf steganography in justified texts". *Journal of Information Security and Applications*, 45, 61-70, 2019.

[41] Doğan S. "A new data hiding method based on chaos embedded genetic algorithm for color image". *Artificial Intelligence Review*, 46(1), 129-143, 2016.

[42] Dogan S. "A reversible data hiding scheme based on graph neighbourhood degree". *Journal of Experimental & Theoretical Artificial Intelligence*, 29(4), 741-753, 2017.

[43] Tuncer T, Avcı D, Avcı E. "İkili imgeler için mayın tarlası oyunu tabanlı yeni bir veri gizleme algoritması". Journal of the Faculty of Engineering and Architecture of Gazi University, 31(4), 951-959, 2016.

[44] Tuncer T. "A probabilistic image authentication method based on chaos". *Multimedia Tools and Applications*, 77(16), 21463-21480, 2018.

[45] Tuncer T, Avci E. "LBP-LSB local binary pattern based data hiding algorithm: LBP-LSB". *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 10(1), 41-47, 2017.

[46] Kıllıoğlu M, Taşkıran M, Kahraman N. "Secure data transmission for multibiometric identity verification systems using steganography and encryption". *Pamukkale University Journal of Engineering Sciences*, 24(2), 173-179, 2018.

[47] Hore A, Ziou D. "Image quality metrics: PSNR vs. SSIM". *20th International Conference on Pattern Recognition*, Istanbul, Turkey, 23-26 August 2010.

[48] Wang Z, Bovik, AC, Sheikh HR, Simoncelli EP. "Image quality assessment: from error visibility to structural similarity". *IEEE Transactions on Image Processing*, 13(4), 600-612, 2004.

[49] Durdu A. "Test Seti ve Matlab Kodları". https://drive.google.com/a/sakarya.edu.tr/file/d/0B-Ku_tMBJJbSSHBfZkpNdW9wM3c/view?usp=sharing (30.12.2020).