T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İMGE İÇERİSİNE LSB EŞLEŞTİRME ALANI TABANLI KAYIPLI İMGE GİZLEYEN YÜKSEK KAPASİTELİ TERSİNİR SIRÖRTME YÖNTEMİ

DOKTORA TEZİ

Ali DURDU

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Tez Danışmanı : Doç. Dr. A. Turan ÖZCERİT

Mayıs 2016

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İMGE İÇERİSİNE LSB EŞLEŞTİRME ALANI TABANLI KAYIPLI İMGE GİZLEYEN YÜKSEK KAPASİTELİ TERSİNİR SIRÖRTME YÖNTEMİ

DOKTORA TEZİ

Ali DURDU

Enstitü Anabilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ

Bu tez 06 /05 /2016 tarihinde aşağıdaki jüri tarafından oybirliği/oyçokluğu ile kabul edilmiştir.

Doç. Dr.
Ahmet Turan ÖZCERİT
Jüri Başkanı

Prof. Dr.
İsmail ERTÜRK
Üye

Doç. Dr.
Ahmet ÖZMEN
Üye

Doç. Dr.
İbrahim ŞAHİN
Üye

Doç. Dr.
Cüneyt BAYILMIŞ
Üye

BEYAN

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Ali DURDU

24.05.2016

ÖNSÖZ

Gün geçtikçe teknolojinin gelişmesiyle yeni nesil bilgisayarlar çoğalmış ve bununla birlikte bilgisayar kullanımı da artmıştır. Tablet bilgisayarların günlük hayatın her yerine girmesi ile bilgisayar kullanımı hem kolaylaşmış, hem de insanların bilgisayar üzerinden iletişimlerinde çok büyük bir artış kaydedilmiştir. İletişimin bu kadar yoğun olduğu günümüzde, güvenli iletişimin önemi artarak devam etmektedir. Kişisel bilgilerin korunması kişisel güvenlik için önemli olduğu gibi, devletlere ve kurumlara ait gizli bilgilerinin korunması da bir o kadar önemlidir. Bu nedenle güvenli iletişimin sağlanabilmesi için birçok yöntem geliştirilmiştir. Günümüzde, gizli bilgilerin 3. kişilerden korunması için şifreleme veya veri gizleme yöntemleri kullanılmaktadır. Bu tezde yeni veri gizleme teknikleri sunulmuş ve deneysel sonuçlarla yöntemin başarımı analiz edilmiştir.

Tez çalışmasında değerli görüşleriyle ve deneyimiyle birçok katkıda bulunarak yardımlarını esirgemeyen danışman hocam Doç.Dr. Ahmet Turan ÖZCERİT'e, tezin her aşamasında hiçbir zaman desteğini eksik etmeyen eşime ve kızlarıma, üzerimde büyük emekleri olan anneme ve babama teşekkür ve şükranlarımı sunarım.

Bu çalışma SAÜ Bilimsel Araştırma Projeleri Komisyonu tarafından desteklenmiştir (Proje No: BAPK-2013-50-02-015).

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

AES : Gelişmiş Şifreleme Standardı (Advanced Encryption Standard)
AVI : Ses Görüntü Birleşimi (Audio Video Interleave)
BMP : Ham Resim Standardı (Bitmap)
CQI : Renkli İmge Kalite Ölçütü (Color Image Quality Measure)
DCT : Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform)
DFT : Ayrık Fourier Dönüşümü (Discrete Fourier Transform)
DWT : Ayrık Dalgacık Dönüşümü (Discrete Wavelet Transform)
EA-1 : 1. Eşleştirme Alanı
EA-2 : 2. Eşleştirme Alanı
EA-3 : 3. Eşleştirme Alanı
EA-4 : 4. Eşleştirme Alanı
EM-1 : Veri gizleme yöntemi 1
EM-2 : Veri gizleme yöntemi 2
EM-3 : Veri gizleme yöntemi 3
EM-4 : Veri gizleme yöntemi 4
EM-5 : Veri gizleme yöntemi 5
EM-6 : Veri gizleme yöntemi 6
HPQA : Histogram Tabanlı Algısal Kalite Ölçütü (Histogram Based Perceptual Quality Assessment)
JPEG : Birleşik Fotoğraf Uzmanları Grubu (Joint Photographic Experts Group)
JPG : JPEG Dosya Formatı Uzantısı
KB : Kilo Bayt (Kilo Byte)
LSB : En Önemsiz Bit (Least Significant Bit)
LSD : En Önemsiz Onluk Sayı (Least Significant Digit)

LSBM : En Önemsiz Bit Eşleştirme Yöntemi (Least Significant Bit Mapping)

MP3 : Film Uzmanlar Grubu Ses Katmanı 3 (MPEG-1 Audio Layer 3)

MPEG : Hareketli Görüntü Uzmanları Birliği (Moving Picture Experts Group)

MSE : Ortalama Karesel Hata (Mean Square Error)

RGB : Kırmızı Yeşil Mavi Renk Tonlaması (Red Green Blue)

PSNR : Tepe Sinyal Gürültü Oranı (Peak Signal to Noise Ratio)

SSIM : Yapısal Benzerlik Kalite Ölçütü (Structural Similarity)

UQI : Evrensel Kalite İndeksi (Universal Quality Index)

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler : Sırörtme, İmge, Veri Gizleme, LSB Eşleştirme

Sırörtme, iletişim dışındaki kişilerin ilk bakıştaki tespitlerini önlemek üzere gizli verinin masum görünen bir taşıyıcı ortam içine gizlenerek iletilmesi yöntemidir. Sırörtmede gizli verinin varlığından iletişim dışındakiler habersizdir. Sırörtme, imge, video, ses veya metin dosyalarına uygulanabilir. Bu tez çalışmasında, renkli imgeler içine renkli imge gizleyen yeni veri gizleme yöntemleri önerilmiştir. Önerilen yöntemlerin ilk üçü tersinir olmayan veri gizleme yöntemleridir. Tersinir olmayan yöntemler örtü imgeyi eşit çerçevelere bölerek mevcut bitlerle eşleştirme alanları oluşturmaktadır. Gizlenecek veri, eşleştirme alanları ile karşılaştırılarak gizlemeye uygun olup olmadıkları tespit edilir. Eşleşme oluştuğunda, gizli verinin yeri işaretlenir. Gizlenen veri, mevcut bitlerle temsil edildiği için sadece işaretleme işleminde bit değişikliği oluşur. Diğer üç yöntem ise tersinir yöntemlerdir. Tersinir yöntemler, 24-bit renkli imge içerisine yüksek kapasitede kayıplı 24-bit renkli imge gizlemektedir. Tersinir yöntemlerde, eşleştirme yöntemine göre değişiklik gösteren eşleştirme alanı aralıkları oluşturulur. Gizlenecek imge eşit parçalara bölünür ve her parçanın girdiği eşleştirme aralığına göre kodlanarak gizleme işlemi yapılır. Önerilen yöntemler, örtü imgede en az değişiklik yaparak en fazla veriyi gizlemeye çalışır. Önerilen yöntemlerin sıraçma algoritmalarına karşı başarımını test etmek için rasgele seçilen 150 imgeye veriler gizlenmiş ve imge bozulma ölçüm yöntemleri ile değerlendirilmiştir. Tez çalışmasında önerilen tersinir yöntemler, literatürdeki yöntemlere göre kapasite olarak iki kat daha verimli, değişiklik analizi olarak %10 daha düşük PSNR değeri elde etmiştir.

A HIGH CAPACITY REVERSIBLE STEGANOGRAPHY METHOD BASED ON LSB MAPPING AREA FOR HIDING LOSSY IMAGES İNTO IMAGES

SUMMARY

Keywords : Steganography, Data Embedding, Image Steganography, LSB Mapping

Steganography is a data hiding method in an innocent media to prevent initial observations from third parties. Thus, third parties are not aware of the presence of the secret data when steganography is concerned. Steganography can be applied to image, video, audio or text files. In this thesis, a set of new data hiding methods have been proposed for still images. The first three are irreversible methods which divide cover media into equal mapping areas to match the bits of secret data. The mapping areas are tested whether they are suitable for hiding the bits of the secret data. If so, the location of the hidden bits are marked. Since the secret data are represented by existing bits in the cover file, cover bits can be changed only during marking procedures. The next proposed three methods are reversible by which high capacity lossy images can be hidden into images. In reversible methods, the mapping areas are created in determined coded ranges. Proposed methods try to embed as much as possible secret data with a minimum change in cover bits. The method proposed have been tested comprehensively against well-known steganalysis algorithms using 150 image files embedded with secret data then the image distortion parameters have been evaluated. The proposed reversible methods have achieved two times better capacity and %10 lower distortion rate as PSNR compared to the studies in the literature.

BÖLÜM 1. GİRİŞ

İnternet teknolojilerinin gün geçtikçe gelişmesi ve hızlı bir şekilde hayatımıza girmesiyle birlikte iletişim olanakları son derece artmıştır. Günlük yaşantımızın bir parçası olan bilgisayarlar sürekli birbirleri ile iletişim halindedirler. Tablet ve akıllı cep telefonlarının yaygınlaşması ile artık insanlar sürekli olarak her yerden internete girmektedir. Bugün her türlü banka, alım satım vs. gibi önemli işlemlerimizi Internet üzerinden yapmamız mümkün hale geldi. Artık makinelerin bile birbirleri ile iletişim içerisinde olduğu bir dünyadayız. Bu kadar yoğun iletişim ortamlarının en büyük sorunu güvenlik olmuştur. Veri transferlerinde bilgiler 3. şahıslar tarafından çeşitli yöntemler ile ele geçirilerek kötü maksatlı kullanılabilir. İletişimde güvenlik unsuru üzerinde yapılmış birçok çalışma vardır. Şifreleme veya sırörtme yöntemleri bunlar arasında sayılabilir.

Şifreleme günümüzde sıkça kullanılan bir güvenlik yöntemidir. Şifrelemede iletilecek veri iletim ortamında üçüncü kişilerin anlayamayacağı şekilde şifreleme algoritmasına bağlı olarak kodlanır. Şifreleme algoritmalarının bulunması ile şifre çözme algoritmaları bulunmuş ve böylece birbirini sürekli geliştiren bir süreç başlamıştır. Şifreleme yönteminde en büyük açık gönderilen verinin anlamsızlığı nedeniyle şifreli olduğunun bilinmesi ve içerdiği bilginin önem taşıdığının anlaşılmasıdır. Bu nedenle şifreli veri çözülemese de iletişimin engellenebilmesi için iletim hattına saldırılarda bulunulabilir.

Şifrelemenin güvenli olmamasıyla yeni yöntemler geliştirilmiş ve veriler farklı şekillerde gizlenerek daha güvenli iletim kanalları geliştirilmiştir. Veri gizleme teknikleri ile veriler iletişim dışındaki kişilerden gizli olarak iletilir. Şifrelemede olduğu gibi veri açıkça gönderilmediği için gizli veri kanalının keşfedilmesi daha güç olacaktır. Veri gizleme teknikleri şifreleme teknikleri ile birleştirilerek daha güvenli

iletişim kanalları oluşturulmaktadır. Aşağıda şifreleme ile sırörtmenin birbirlerine göre üstünlük ve zayıflıkları verilmiştir.

Şifrelemenin üstünlükleri;

1. Şifreleme kırılmadığı sürece güvenli iletişimi sağlar.
2. Sırörtmeye göre daha yaygın kullanılmaktadır.
3. Kurumsal uygulamalarda vazgeçilmezdir.

Şifrelemenin zayıflıkları;

1. Gizlediği bilgiyi açıkca teşhir ederek cazibe uyandırır. Bu sebeble saldırılara maruz kalmaktadır.
2. İletişimde kullanılan anahtarların herkese açık güvensiz ortamda paylaşılmasının gerekliliği
3. Her şifrenin kırılmaya açık olması
4. Şifrelenmiş verilerin güvenliğini güncel tutabilmek için belirli periyodlarla şifrelerin güncellenmesinin gerekliliği

Sırörtmenin üstünlükleri;

1. Gizlenen veri şifrelemede olduğu gibi açıkça teşhir edilmez. Gizli veri masum görünen ortama gizlenerek güvenli iletişim habersizce sağlanır.
2. Şifrelemeye göre en önemli üstünlüğü tespit edilmesi çok daha zordur.
3. Sıraçma yöntemleri gizli verinin varlığını tespit edebilseler bile gizli veriye erişecek düzeye gelememiştir.

Sırörtmenin zayıflıkları;

1. Şifrelemeye göre daha az yaygındır.
2. Sırörtme münferit uygulamalarda kullanımı daha uygundur.

Şifrelemenin yaygınca kullanılmasının yanında pek çok zayıflığı bulunmaktadır. Sırörtme şifrelemeye göre daha az yaygın olmasına rağmen pek çok üstünlüğü bulunmaktadır.

Veri gizleme tekniklerinin tersine çalışan gizli veri analiz çalışmalarına da sıraçma adı verilmektedir. Veri gizleme teknikleri kötü niyetli kişiler tarafından kullanımı büyük zararlarla sonuçlanabilir. Bunun önüne geçilebilmesi için sıraçma çalışmaları ile ortamdaki gizli bilginin varlığı tespit edilebilir. Sıraçma çalışmalarının temel çalışma mantığında veri gizleme işlemlerinin izlerini sürmek vardır. Her veri gizleme işlemi birtakım parmak izleri bırakmaktadır. Bunu tesbit eden sıraçma çalışmaları ortamdaki gizli verinin varlığını sezebilir.

Sırörtme sadece güvenlik için değil iletişimin gizli olarak yapılması gerekli durumlarda da çok kullanışlı bir yöntemdir. Örneğin devletlerin gizli bilgileri ilgili kişilerle habersizce ulaştırabilmesi buna bir örnek olarak verilebilir. Sırörtme resim, ses ve video ortamlarına uygulanabilir.

1.1. Tez Konusu İle İlgili Literatürdeki Çalışmalar

Sırörtme ile ilgili literatürde birçok çalışma mevcuttur. Bunlardan en yaygın kullanılan teknik LSB yöntemidir. LSB yöntemini baz alarak geliştirilmiş eşleştirme yöntemleri örtü ortamda daha az değişiklik yapmayı hedeflemektedir. Literatürde eşleştirme yöntemi ile veri gizleyen çalışmalar mevcuttur.

Eşleştirme yöntemini ilk kez kullanan Sharp [1], LSB yer değiştirme metodundan farklı olarak eğer örtü dosyanın veri gizlenecek baytının son biti gizlenecek bit ile aynı ise ilgili baytın son bitinde herhangi bir değişiklik yapılmaz. Eğer örtü dosyanın sıradaki baytının son biti gizlenecek bitden farklı ise o zaman örtü dosyanın bayt değeri 1 arttırılır veya 1 azaltılır. Bu şekilde gizlenmek istenen bit bilgisi son bite gizlemiş olur. Buradaki amaç örtü dosyada daha az değişiklik yaparak gizlenen bilginin sıraçma algoritmaları tarafından tespit edilmesini zorlaştırmaktır.

Mielikainen yaptığı çalışmada durağan imgeye eşleştirme yöntemi kullanarak veri gizlemiştir. Önerilen yöntem bir fonksiyon yardımıyla veri gizlemektedir. Yöntem ardışık iki piksele 2-bit veri gizleyerek çalışır. Yöntemde ilk pikselin LSB biti gizlenmiş 2-bitlik mesaj parçasının 1. bitini ardışık iki pikselin en önemsiz bitlerinin fonksiyon sonucu ise mesaj parçasının 2.bitini gösterir. Yöntem imgede en az değişiklikle her piksele bir bit veri gizlemektedir. Bu yön0temle 8 bit veri gizlendiğinde 3-bitlik bir değişim olmaktadır [2].

Chan, Mielikainen'in yaptığı çalışmayı geliştirerek yeni bir yöntem önermiştir. Chan'in önerdiği yöntem fonksiyon olarak XOR kapısını kullanmaktadır. Yöntem Mielkainen'in önerdiği yöntem gibi ardışık iki piksele 2-bit veri gizlemektedir. Yöntem imgedeki ardışık iki pikselden ilk pikselin LSB biti ile ikinci pikselin LSB bitinden bir önceki biti XOR alarak veri gizlemektedir. Bu yöntemle her bir piksele bir bit gizlenerek örtü imgede daha az değişiklik yapmayı hedeflemektedir. Gizlenecek bitler örtü imgeye sıralı olarak gizlenir [3].

Tian örtü imgede düşük bozulumlu, yüksek kapasiteli ve tersinir veri gizleme yöntemi önermiştir. Tian, önerdiği yöntemde örtü imgenin iki pikseli arasındaki farkı iki kat genişleterek oluşan bölgeye veri gizledi [4].

Alatlar, Tian'ın önerdiği yöntemi geliştirmiş ve dört piksel arasındaki farkı iki kat genişleterek 3-bitlik veriyi oluşan bölgeye gizledi. Alatlar Tian'ın önerdiği yönteme göre daha fazla veri kapasitesi sundu [5].

Chang ve arkadaşları yaptıkları çalışmalarında örtü imgeyi iki kez oluşturdular. Oluşan iki imgeye modül matrisi ve değişiklik yönünü kullanarak veri gizlediler. İki imge olduğu için yüksek veri kapasitesi sundular [6].

Lu ve arkadaşları Chang'in[6] önerdiği yöntemi geliştirerek alternatif bir yöntem önerdiler [7]. Önerilen yöntem, kamuflaj pikselleri ile örtü imgelerde yüksek görüntü kalitesinin yanında yüksek veri gizleme kapasitesi sunmaktadır.

Ker yaptığı çalışmada 2/3 verimli gizleme yöntemi önermişlerdir. Bu yöntem iki bit veri gizlemek için üç piksel kullanır. İki pikselin son biti veri gizleme amacında kullanılırken son piksel bilginin aynı şekilmi veya tümleyeninin mi gizlendiğini gösterir. Eğer gizlenecek iki bit, tümleyeni alınarak gizlendiğinde örtü imgenin son bitlerinde daha az değişiklik yapıyorsa tümleyen olarak, değilse değişiklik yapılmadan gizlenir. Böylece son bitlerde en az değişiklik yaparak veri gizleme çalışır [8].

Wu ve Tsai piksel farkı yöntemi ile veri gizleme yöntemini önermişlerdir Bu yöntemde örtü 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 [9].

Wang ve arkadaşları Wu ve Tsai'nin[9] yaptığı çalışmayı geliştirerek yeni bir yöntem önermişlerdir. Yeni yönteme iki piksel arasındaki farkın modül fonksiyonu sonucunu kullanarak veri gizleme esasına göre çalışmaktadır [10].

Fridrich ve Soukal hamming matrisini kullanarak veri gizleme yöntemi önermişlerdir [11]. Yöntem örtü imgede diğer yöntemlere göre daha az değişiklik yapmaktadır. Ayrıca yöntemin yüksek veri kapasitesi ile veri gizleme özelliği de vardır.

Kurtuldu ve Arıca imge kareleri yöntemi adını verdikleri çalışmalarında yeni bir veri gizleme yöntemi önermişleridir. Önerdikleri yöntem, örtü imgeyi bloklara bölerek gizlenecek mesaj bilgisini blok içerisindeki piksellerin son bitlerinin dizilimlerine bakarak mesaj bilgisine en yakın dizilime sahip piksel gurubunu gizleme işlemi için seçer. Bu yöntemde örtü imgeye gizlenen bilginin kapasitesi düşüktür [12].

Soleimanpour-Moghadam ve Nezamabadi-pour yaptıkları çalışmalarında ayrık kuantum davranışlı ağırlıklı arama algoritması geliştirmişlerdir. LSB eşleştirme yönteminde üç skorlu puanlama sistemi kullanmışlardır. Geliştirdikleri arama

algoritması ile en iyi permütasyon sıralamasını bularak veri gizlemişlerdir. Mielkainen'in yöntemi ile çalışmalarının performansını kıyaslamışlardır [13].

Wu ve arkadaşları çalışmalarında sırlı imgeyi iki kez oluşturarak ilk sırlı imgeye veriyi, ikinci sırlı imgeye ise gizlenen verinin referans bilgilerini gizler. DH anahtarı olmadan sırlı imgelerden gizli veri çıkarılamaz. [14]

Huang ve arkadaşları kenar uyarlamalı bitişik piksel çifti eşleştirme yöntemi ile veri gizlemişlerdir. Bitişik piksel çifti seçiminde yeni bir rasgele seçim yöntemi önermişlerdir [15].

Sabeti ve arkadaşları yaptıkları çalışmalarında, imgedeki veri gizlemede güvenli bölgeyi belirlemek için karmaşıklık ölçütü kullanmaktadırlar. Çok sayıda saldırılara karşı geleneksel eşleştirme yöntemlerine göre daha güvenli olduğu göstermişlerdir [16].

Jain ve Kumar kayıpsız veri sıkıştırması ile imge içine verimli veri gizleme yöntemini önermişlerdir. Jain ve Kumar yaptıkları çalışmalarında gizli veriyi kayıpsız veri sıkıştırmasıyla gizlemişlerdir. Önerilen yöntem yüksek veri kapasitesi sağlamaktadır [17].

Tablo 1.1.'de eşleştirme yöntemi ile ilgili literatürdeki sırörtme çalışmaları özetlenmiştir.

Tablo 1.2.'de literatürdeki 5 yöntemin 256x256 boyutlarındaki örtü imgeye 100 bayt verinin gizlenmesiyle oluşan sırlı imgelerin PSNR değerleri, örtü imgenin son bitlerindeki değişen bit sayıları ve örtü imgenin son bitlerini (LSB) değişim oranları (Değişen Bit/Gizlenen Bit) verilmiştir.

Tablo 1.1. Eşleştirme yöntemine göre sırörtme ile ilgili literatürdeki çalışmalar

Yazar	Özellikler
Sharp[1]	- Gizlenecek bit ile örtü biti aynıysa örtü biti değiştirmez - Gizlenecek bit ile örtü biti farklıysa örtü pikseli 1 arttırır veya 1 azaltır.
Mielikainen[2]	- Örtü piksellerini piksel çifti olarak kullanır - Piksel çiftlerinden ilk pikselin son bitine 1 bit gizlenir. - Piksel çiftinin fonksiyon sonucuylada 1 bit gizlenmiş olur
Chan[3]	- Mielikainen'in[2] önerdiği yöntemi geliştirmiştir. - Fonksiyon olarak xor yöntemini kullanır.
Tian[4]	- İki piksel arasındaki farkı iki kat genişleterek veri gizlemiştir
Alattar[5]	- Tian'ın[4] yöntemini geliştirmiştir. - Dört piksel arasındaki farkı iki kat genişleterek 3bitlik bilgiyi gizlemiştir.
Chang ve ark.[6]	- Örtü imgeyi ik kez oluşturdular. - Modül matrisi ve değişiklik yönünü kullanarak veri gizlediler. - Örtü imgeyi iki kez oluşturarak veri gizleme kapasitesini arttırmışlardır.
Lu ve ark.[7]	- Chang'in yöntemini değiştirerek alternatif bir yöntem önerdiler.
Ker[8]	- 2-bit veri gizlemek için 3 piksel kullanır - İki piksel veri gizleme amacında kullanılırken son piksel gizlenen verinin formatını gösterir Son piksel gizlenen verinin tümleyen olarak gizlenip gizlenmediği gösterir
Wu ve Tsai[9]	- Ardışık piksellerin farklarını sınıflandırarak veri gizler
Wang ve ark.[10]	- Wu ve Tsai'nin[9] yaptığı çalışmanın geliştirilmişidir - İki piksel farkının modülasyon fonsiyonu sonucu veri gizler.
Fridrich ve Soukal[11]	- Gizleme yapabilmek için hamming matrisini kullanmışlardır
Kurtuldu [12]	- İmgeyi bloklara bölerek gizlenecek bit dizisini blok içerisindeki piksellerin son bitlerinin dizilişinde arayıp en yakın olanına gizlemektedir.

Tablo 1.2. 256x256 imgeye literatürdeki çalışmaların önerdiği yöntemlerin 100 bayt veri gizlemeleriyle oluşan sırlı imgelerin PSNR değerleri ve değişen piksel sayıları [18].

Yöntem	PSNR	Değişen Bit Sayısı	Değişim Oranı
Chan [3]	72.23	255	%30
Mielikainen [2]	71.73	286	%35
LSB	70.63	368	%46
Wang ve ark [10].	67.30	400	%75
Wu ve Tsai[9]	62.51	434	%86

1.2. Tezin Motivasyonu

İyi bir sırörtme yönteminin veri gizlerken örtü ortamda en az değişiklik yapması beklenir. Literatürde birçok sırörtme yöntemi mevcuttur. Literatürdeki birçok makale örtü ortamda daha az değişiklik yapmak için yeni yöntemler önermektedir. Tablo 1.2.'de literatürdeki 5 çalışmanın 256x256 boyutlarındaki örtü imgeye 100 bayt verinin gizlenmesiyle oluşan sırlı imgelerin PSNR değerleri, örtü imgenin son bitlerindeki değişen bit sayıları ve örtü imgenin son bitlerini (LSB) değiştirme oranları verilmiştir. Bu tablodan da anlaşılacağı üzere literatürdeki çalışmalar öncelikle olarak örtü imgede daha az nasıl değişiklik yapılabileceğini araştırmaktadır. Değişiklik sayısını azaltırken veri gizleme kapasitesi de ihmal edilmemelidir. 24-bit renkli imgeler için eşleştirme alanı tabanlı yüksek kapasiteli düşük değişiklikli kayıplı 24-bit renkli imge gizleyen yeni bir sırörtme yaklaşımının geliştirilmesi tez çalışmasının motivasyonunu oluşturmaktadır.

Bu tez çalışmasında yeni bir veri gizleme yaklaşımı önerilmiş ve bu yaklaşımdan türetilmiş 6 adet yöntem sunulmuştur. Önerilen yaklaşımla 4-bit ile gizlenen 4-bit veri için 1 bitte değişiklik oluşmaktadır. Böylece 1-bit değiştirilerek 4-bitlik veri saklanabilmektedir. Geleneksel sırörtme yönteminin (LSB) veri gizlemesinde örtü ortamda %50 oranında değişim oluşmaktayken, önerilen yöntemin %25'lik değişim oranı Tablo 1.2.'de verilen literatürdeki çalışmaların değişim oranlarına göre daha düşüktür.

1.3. Tezin Amacı ve Katkıları

Bu tezin amacı; örtü imgede %25 değişiklikle 2 bpp oranına kadar yüksek kapasiteli 24-bit renkli kayıplı imge gizleyebilen yeni bir sırörtme yöntemi önermektir. Eşleştirme yaklaşımıyla geliştirilen veri gizleme yöntemlerinde amaç, gizlenecek veriyi örtü ortamın bitlerini değiştirmeden yerleştirmek ve mümkün olabilecek en az değişiklikle veri gizleme işlemini tamamlamaktır. Tezde önerilen yöntemler eşleştirme yaklaşımı esas alınarak geliştirilmiştir.

Tez çalışmasının iki ana katkısı bulunmaktadır :

1. Örtü imgede %25 oranında değişiklikle veri gizleyen sırörtme yöntemini önermek.
2. Örtü imgeye 2 bpp oranında yüksek kapasiteli imge içine imge gizleyen sırörtme yöntemini önermek.

1.4. Tez Organizasyonu

Geliştirilen sırörtme yöntemi ve değerlendirilmesini içeren bu tez, beş ana bölümden oluşmaktadır.

Birinci bölümde giriş, sırörtme teknikleri üzerine yapılan çalışmalar, tezin amacı, motivasyonu, katkıları ve organizasyonu sunulmaktadır.

İkinci bölümde sırörtme tekniklerinin çeşitleri ve detayları üzerinde durulmaktadır.
Üçüncü bölümde tez çalışmasında geliştirilen yeni sırörtme yöntemlerinin detayları anlatılmaktadır.

Geliştirilecen yöntemlerin karşılaştırmalı değerlendirmesi dördüncü bölümde sunulmaktadır.

Beşinci bölümde sonuçlar özetlenerek ve gelecek çalışmalar irdelenmektedir.

BÖLÜM 2. SIRÖRTME KAVRAMI VE YÖNTEMLERİ

Gün geçtikçe teknolojinin gelişmesiyle yeni nesil bilgisayarlar çoğalmış ve bununla birlikte bilgisayar kullanımı da artmıştır. Tablet bilgisayarların ve akıllı telefonların günlük hayatın her yerine girmesi ile bilgisayar kullanımı hem kolaylaşmış hem de insanların bilgisayar üzerinden iletişimlerinde büyük gelişmeler kaydedilmiştir. Facebook, Twitter, Instagram ve Whatsapp gibi birçok uygulama ile insanlar gün içerisinde sürekli haberleşmektedirler. Ayrıca bankacılık, alışveriş, ödeme işlemleri gibi tüm önemli işlemler İnternet üzerinden yapılmaktadır.

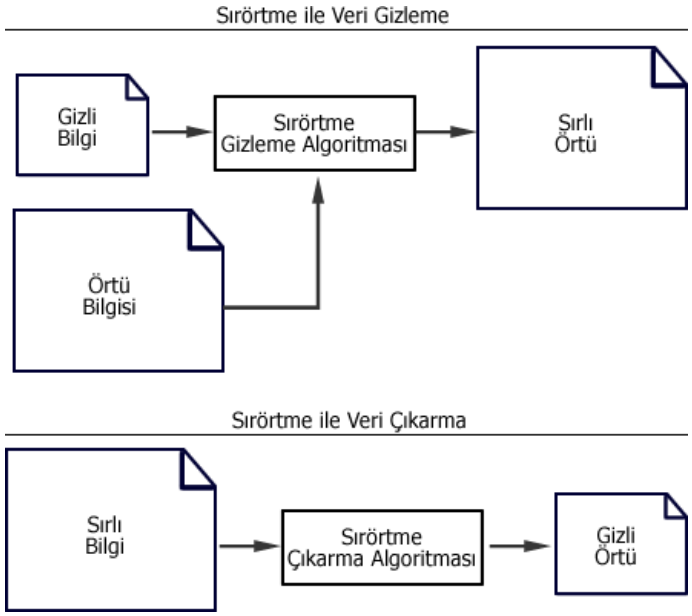
İletişimin bu kadar yoğun olduğu günümüzde güvenli iletişim konusu da güncelliğini korumaktadır. Güvenli iletişim, kişisel güvenlik için önemli olduğu gibi devlet kurumlarına ait bilgilerin de güvenliği içinde önemlidir. Bu nedenle güvenli iletişimin sağlanabilmesi için birçok yöntem geliştirilmiştir.

Şifrelemenin yanında sırörtme teknikleri de sıklıkla kullanılmaktadır. Veri gizleme kavramı, çok eskilere dayanmaktadır. Sırörtmenin geçmişi Antik Yunan medeniyeti zamanına kadar dayanmaktadır. Sırörtme kelimesinin kökleri "στεγαυος" ve "γραΦειν"'den gelen Yunan alfabesinden türetilmiştir [19]. Kelime anlamı olarak gizli yazı veya örtülü yazı anlamına gelmektedir [19]. Amacı gizli iletişime olanak sağlamaktır. Geçmişte sırtörtmenin uygulanması ile ilgili İran savaşları sırasında Herodot, kafasını kazıtarak mesaj yazdırmaya izin veren bir ulağı anlatır. Mesaj yazıldıktan sonra ulağın saçlarının uzaması beklenir. Ulak gideceği yere giderek kafasını kazıtır ve gizli mesajı iletmiş olur. Bu, sırörtmenin tarihte ilk kullanımıdır [20]. Yine Heradot'un bildirdiğine göre, M.Ö. 440 yılında Demaratus'un Yunanistan'a karşı bir saldırı tehlikesini bildirmek için, gizli mesajı tahta bir tabletin üzerine kazıdıktan sonra üzerini balmumu ile kaplamasıdır. Üzerinde balmumu

bulunan tablet hiçbir kuşku uyandırmazken, ısıtılıp mum eritildiğinde gizli mesaj ortaya çıkmaktadır [20].

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 [21]. 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. LSB tekniğinin pek çok zayıflıkları vardır. Örneğin taşıyıcı dosyaya gürültü eklenmesi ile LSB tekniği ile gizlenen veri yok edilebilir [20].



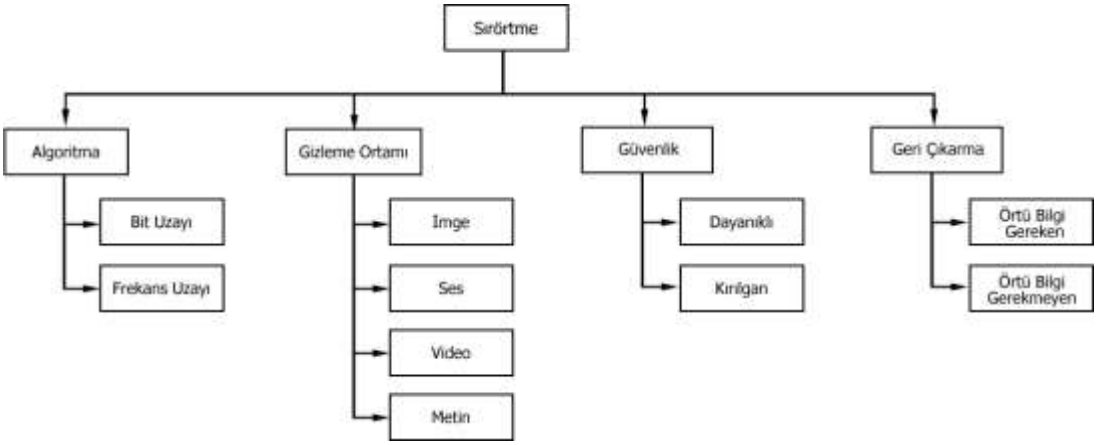
Şekil 2.1. Sırörtme ile veri gizleme ve veri çıkarma işleminin genel diyagramı

Şekil 2.1.'de sırörtme ile veri gizleme ve veri çıkarma işleminin genel diyagramı verilmiştir. Diyagrama göre gizli veri örtü bilgisine gizleme algoritmasıyla gizlenerek sırlı örtü elde edilir. Sırlı örtü, örtü ile aynı formdadır. İnsan gözü veya işitme sistemi tarafından fark edilmez. Sırlı veri karşı tarafa iletildikten sonra karşı

taraf çıkartma algoritmasını kullanarak gizli bilgiyi elde eder. Bu şekilde masum görünen bir dosya gizli bilgileri taşıyabilir.

2.1. Sırörtmenin Sınıflandırılması

Sırörtme, kullanılan algoritmaya göre, gizli mesajın gizlendiği veri ortamına göre ve güvenliğe göre üç ana başlık altında sınıflandırılabilir. Şekil 2.2.'de sırörtmenin sınıflandırma şemasını görülmektedir.



Şekil 2.2. Sırörtmenin sınıflandırılması [22]

2.2. Algoritmaya Göre Sırörtme Teknikleri

Sırörtme, algoritmaya göre bit uzayı ve frekans uzayı adı altında iki teknikle uygulanmaktadır. Bit uzayı tekniğinde, geleneksel en önemsiz bite (LSB) veri gizleme yöntemi kullanılır. Bu en yaygın kullanılan yöntemdir. Frekans uzayı tekniğinde ise dönüşümler kullanılarak katsayılara matematiksel olarak gizleme yapılır. Bu yöntem karmaşık ve geleneksel LSB yöntemine göre daha az kullanılan bir yöntemdir.

2.2.1. Bit uzayı

Sırörtme teknikleri arasında en yaygın kullanılan yöntem bit uzayı yöntemleridir. Bu yöntemlerden en yaygın kullanılanı LSB en önemsiz bit yöntemidir. Bu yöntemde, veri gizlerken örtü dosya ve gizlenecek veri ikili sisteme çevrilerek işlem yapılır. Örtü dosyasındaki her bir baytlık verinin son biti en önemsiz bit olduğu için değiştirildiğinde dosyada algılanacak bir değişikliğe neden olmaz. Bu durumdan yararlanarak tasarlanan LSB yönteminde son bitler veri gizlemek amacıyla kullanılır. Gizlenecek mesajın bitleri masum görünen örtü dosyanın son bitlerine gizlenir. Gizli bilgiyi taşıyan örtü dosya karşı tarafa kimsenin şüphesini çekmeden gönderilir.

LSB yönteminin yaygın olmasının nedenlerinden birisi basit olmasıdır. Özellikle kaliteli resimlerde yüksek veri gizleme kapasitesi sunar. Yüksek kaliteli dosyalar internet üzerinden aktarılırken boyutu nedeniyle zorluk çıkarabilir. Yüksek boyutlu resimlerin içine LSB yöntemiyle veri gizlemesinden sonra sıkıştırma işlemine tabi tutulması durumunda veri kaybı olduğundan gizlenen verinin bilgileri kaybolur. Bu nedenle küçük boyutlu resimlerin tercih edilmesi gereklidir.

Bit uzayını kullanan bir diğer yöntem ise eşleştirme yöntemidir. Eşleştirme yönteminde LSB yönteminde olduğu gibi son bitler doğrudan değiştirilmez. Bunun yerine en önemsiz bit gizlenecek bit ile aynı ise bitte değişiklik yapılmaz, aynı değilse ilgili bayt verisi 1 arttırılır veya 1 azaltılır. Eşleştirme yöntemi ilk kez Sharp tarafından önerilmiştir [1]. Bu yöntem frekans uzayında kullanıldığında istatistiksel saldırılara karşı daha dayanıklı hale getirilmiştir [23].

Mielikainen, Sharp'ın önerdiği yöntemi geliştirerek gizleme işlemini fonksiyonun verdiği sonuca göre yapmaktadır [2]. Yönteme göre örtü imge piksel çiftlerine ayrılmıştır. Gizli veri iki bitlik guruplar halinde gizlenmektedir. Gizlenecek ilk bit ilk pikselin son bitine doğrudan gizlenmektedir. İkinci bit ise piksel çiftinin son bitlerinin gizleme fonksiyon sonucuyla elde edilir. Fonksiyon sonucunun ikinci gizleme bitini verebilmesi için piksel çiftlerinden birisi 1 arttırılır veya 1 azaltılır.

Mielikainen'in yöntemi LSB yöntemine göre örtü imgede çok daha az değişiklik yapmaktadır. Bunun yanında LSB yöntemiyle aynı kapasitede veri gizlemektedir.

Akar'ın önerdiği LSD adlı yöntemin örtü imgedeki bayt verileri onluk sisteme çevrilerek birler basamağını değiştirme mantığıyla çalışmaktadır [19]. Örneğin örtü imgenin bayt verisi 250 ise ve gizlenmek istenen veri 4 ise örtü imgenin değişmiş bayt verisi 254 olur. Bu yönteme göre gizleme işleminde her bir bayt için 3-4 bit arası değişiklik yapılmaktadır. Yöntemin dezavantajı örtü imgede LSB yöntemine göre oldukça fazla değişiklik yapmaktadır. Bunun yanında veri kapasitesi LSB yöntemine göre 3 kat fazladır.

2.2.2. Frekans uzayı

Frekans uzayında örtü dosyaya veri gizlemek en karmaşık yöntemdir. Bu nedenle bit uzayı yöntemlerine göre çok yaygın değildir. Bunlardan bazıları Ayrık kosinüs dönüşümü (DCT), ayrık dalgacık dönüşümü (DWT) ve ayrık fourier dönüşümüdür (DFT). Bu yöntemler ile veri gizlerken ilk olarak örtü dosyaya kullanılan dönüşüm yöntemi uygulanır. Dönüşüm sonucunda oluşan bileşenlerin katsayı çarpanlarına bakılarak veri gizlenecek bölgeler tespit edilir. Daha sonra veri gizlenebilecek bölgelerinin dönüşümlerinin katsayılarında değişiklik yaparak veri gizleme işlemi tamamlanır. [22]. Katsayı eğer sıfır değerinde ise bu katsayıda yapılan değişiklik insan göz sistemi tarafından algılanamaz. Bu katsayılarda değişiklik yapılarak veri gizleme işlemi yapılır. Frekans uzayında gizlenen veriler sıraçma yöntemleri tarafından daha zor tespit edilir ve yöntemin ataklara karşı dayanıklılığı daha yüksektir. Dönüşüm yöntemiyle frekans uzayında veri gizleme yöntemlerinin birçoğu tüm veri gizleme ortamlarını desteklemektedir. Bu yöntemle gizlenen verilerin silinmesi imkânsızdır. Sırlı dosyaya yapılacak görsel saldırılardan etkilenmez [23].

Jpeg ve MPEG dosya formatı DCT dönüşümünü kullanmaktadır. Jpeg ve MPEG formatları bilinen en iyi sıkıştırma yöntemlerindendir [23].

Ruanaidh ve arkadaşları 1996 yılında yaptıkları çalışmalarında gizli veriyi DFT fourier dönüşümündeki faz bileşenlerin algısal olarak önemli olanların katsayılarına gizlemişlerdir. Önerilen yöntemde büyük genliğe sahip dönüşüm katsayıları tercih edilmiştir [25].

Podilchuk ve Zeng yaptıkları çalışmalarında görüntü uyarlamalı DCT blok bazlı veri gizlemeyi önermişlerdir [26]. Hernández ve ark. genelleştirilmiş Gauss yaklaşım ile veri gizlemeyi önermişlerdir [27]. Çalışmalarında DCT yöntemiyle veri gizleme işleminin teorik özelliklerini inceleyerek gerçeğe uyarlayarak sistem parametrelerini kontrol etmişlerdir. Bu sonuçlardan yararlanarak en uygun algılama eşiğini seçmişlerdir.

2.3. Veri Ortamına Göre Sırörtme

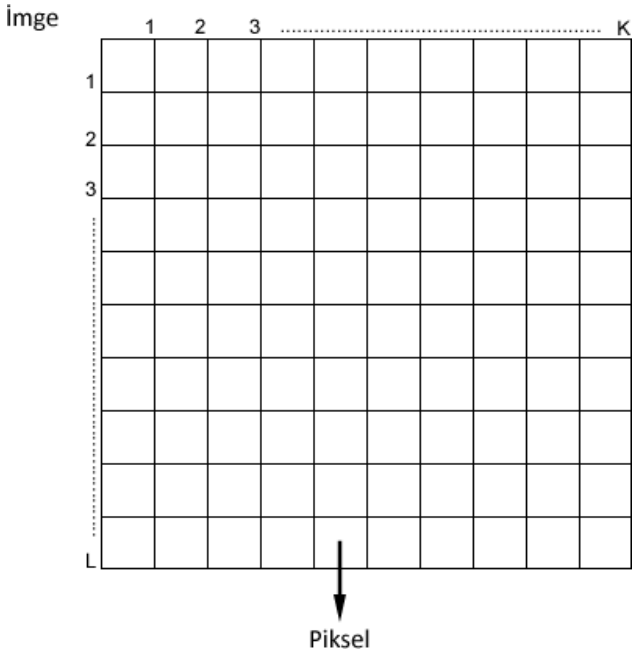
Sırörtme bilimi eski çağlarda kafa derisine kazınan mesaj olarak uygulandığı gibi tabletlere yazılan yazının balmumuyla kaplanarak gizlenmesi şeklinde de uygulanmıştır. Bilgisayar dünyasının çok geliştiği günümüzde sayısal ortamlara da uygulanmaktadır. Günümüzde sırörtme tüm dosya ortamlarına uygulanabilmektedir. Bu bölümde sırörtmenin uygulandığı ortamlar incelenecektir.

2.3.1. İmge dosyalarında sırörtme

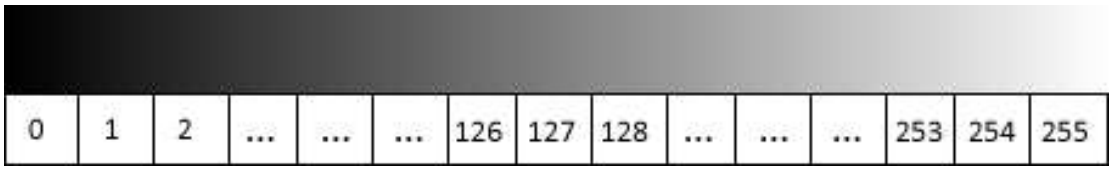
Sırörtmenin bilgisayar dünyasında en çok uygulandığı ortam imgelerdir. Literatürdeki sırörtme çalışmalarının birçoğu imgeler için önerilmektedir. Günümüzde internet ve bilgisayar teknolojilerinin çok fazla gelişmesiyle resimler internet üzerinden günün her anında paylaşılmaktadır. Özellikle sosyal medya ortamlarında insanlar her anlarını paylaşım yaparak geçirmektedirler. Bu kadar yoğun resim paylaşımının olduğu ortamda güvenlik önemli bir yer edinmektedir.

İmge dosyalarına gizli metin gizlenebileceği gibi bir imgenin içine başka imge ya da video gizlemekde mümkündür. İmgenin içerisine dosya formatından bağımsız olacak

şekilde her türden dosya gizlemek mümkündür. Burada önemli olan imgenin alabileceği veri miktarıdır. Geleneksel LSB yönteminde bir imge boyutunun 8 de 1 kadar veri saklayabilir. İmgelerin en küçük yapı taşına piksel adı verilir. Her bir imge, piksellerin bir araya gelmesiyle oluşur. Şekil 2.3.'de gösterildigi gibi sayısal imgeler KxL'lik bir matris olarak gösterilebilir. 1..K arası sütunları, 1..L arası ise satırları göstermektedir. Gri seviye siyah beyaz imgelerde her bir piksel 1 bayt ile temsil edilir. 1 baytlık veri 0-255 arasında değere sahip olabilir [28]. Şekil 2.4.'de de görüldügü gibi 0 ile 255 arası tüm değerler gri seviyesindedir. Fakat en uç noktalar da durum farklıdır. 0 siyah 255 ise beyazdır. Bu iki değer arasındaki tüm renkler gridir.



Şekil 2.3. Örnek bir imgenin piksellerinin gösterilişi



Şekil 2.4. Gri seviye (gray level) imgelerin renk paleti

Renkli imgelerde her bir piksel 3-baytlık veri ile temsil edilir. Renkli imgelerde RGB (Red-Green-Blue) adı verilen renk kodlama sistemi kullanılmaktadır. Her pikselde üç

ayrı renk verisi bulunmaktadır. Bu renk verilerinin değerlerine göre birleşimi ile farklı bir renk oluşur. Bu üç ana renkte sayısal değerler küçüldükçe oluşan renk daha koyulaşmaktadır. RGB renk modelinde 256 x 256 x 256 = 16.777.216 adet farklı tonda renk oluşturulabilir. Böylece belirli bir pozisyondaki pikselin Tablo 2.1.'de görüldüğü resmin bileşenlerinin şiddetini belirler.

Tablo 2.1. RGB renk tablosu

Renk		R	G	B
	Kırmızı	255	0	0
	Yeşil	0	255	0
	Mavi	0	0	255
	Turuncu	255	153	0

Tablo 2.1.'de görüldüğü gibi kırmızı rengini oluşturan renk tonlamasında kırmızı (R) renk 255, yeşil (G) renk 0 ve mavi (B) renk de 0 değerine sahiptir. Aynı şekilde yeşil rengini oluştururan renk tonlamasında R verisi 0, G verisi 255, B verisi ise 0 değerine sahiptir. Ara renk olan turuncu rengini oluşturan renk tonlamasına göre kırmızı (R) verisi maksimum, yeşil (G) verisi 153 ve mavi (B) verisi 0 değerine sahiptir. Yani turuncu rengi, saf kırmızı ile orta değerde yeşil karışımıyla oluşmuştur. Böylece üç ana rengi karıştırarak istenilen tonda renk elde edilebilir.

2.3.2. Ses dosyalarında sırörtme

İnternetin yoğun kullanıldığı günümüzde ses dosyalarının paylaşımı da artmıştır. Ses dosyalarının boyutları veri gizlemeye oldukça uygundur. Veri gizleme çalışmaları imgelerden sonra 2000'li yıllardan itibaren ses dosyaları içinde yapılmış ve ses dosyalarına veri gizlemeye rağbet artmıştır. Resim dosyalarına benzer şekilde ses içerisine veri saklama yöntemleri de insan işitme sisteminin zafiyetinden yararlanılarak geliştirilmiştir. Fakat insan işitme sistemi göz sistemine göre daha hassastır. Bu nedenle yapılacak en küçük değişiklik bile algılanabilir. Bu yüzden ses dosyalarında yapılan veri gizleme teknikleri imgelerde kullanılan veri gizleme tekniklerine göre daha karmaşıktır [29], [30].

Ses dosyalarının farklı formatlarına veri gizlenebilir. Veri gizlemede en fazla kullanılan ses formatları WAV ve MP3 formatlarıdır. Yapılan çalışmalar ham veri içermesinden dolayı işlenmesinin kolay olduğu WAV dosyalarında yoğunlaşmıştır. MP3 ses dosyalarına veri gizlemeden önce sıkıştırılmış dosyanın açılması ve ham verilerin elde edilmesi gibi ara işlemler vardır. Bununla birlikte MP3 dosyaları, WAV dosyalarında göre daha fazla yaygın bir şekilde kullanılmaktadır. Boyutlarını küçük olması dolayısıyla internet üzerinde paylaşımı kolaydır. Mp3 dosyalarına veri gizleyen uygulamalardan en bilineni Mp3Stego yazılımıdır [31]. Bu yazılım sıkıştırma esnasında MP3 dosyaları içerisine veri saklayabilmektedir.

Ses içerisine veri gizleme yöntemleri: aşama kodlaması (phase coding), en düşük değerli bit kodlaması (LSB), yankı veri saklaması (echo data hiding) ve tayf yayılması (spread spectrum) olarak sınıflandırılmaktadır [29].

2.3.2.1. Aşama kodlaması

Ses dosyalarına veri gizleme yöntemlerinden aşama kodlaması yönteminde, ses dosyası küçük segmentlere bölünür. Oluşan segmentlerin fazları gizlenecek verinin fazları ile değiştirilir [29].

2.3.2.2. En düşük değerli bit

İmgeler için uygulanan LSB yöntemi ses dosyaları için de uygulanabilir. En düşük değerli bit kodlaması ses örneklerinin son bitleri ile gizlenecek verinin son bitlerini değiştirme işlemidir. İşlem sonrasında oluşan gürültüler ses dosyasında bozulmalara neden olur. Bu bozulmalar insan işitme sisteminin algılayacağı seviyelere gelebilir. Ayrıca sıraçma saldırılarına karşı dayanıksız bir yapısı vardır [29], [32].

2.3.2.3. Yankı

Ses dosyalarındaki ses sinyali üzerine yankı sesi eklenerek ve yankının farklı gecikme değerleri kodlanarak veri saklanabilmektedir. İnsan kulağı ses örneklerinde milisaniye zaman dilimindeki değişiklikleri algılayamaz. Ses örneklerine gizli bilgiler içeren milisaniye uzunluğunda yankılar eklenir [33].

2.3.2.4. Tayf yayılması

Ses dosyalarında tayf yayılması ile veri gizleme tekniğinde gizli veriler ses örneklerinin frekanslarının tayflarına gizlenmektedir. Ses örneklerinde istenmeyen gürültü sesleri oluşturabilir. Bu ise en büyük dezavantajıdır [29].

2.3.3. Video dosyalarında sırörtme

Videolar hareketsiz görüntülerin arka arkaya belirli süre aralıkla gösterilmesiyle oluşur. Aynı şekilde arka plandaki ses de senkron bir şekilde çalmaktadır. Videoyu oluşturan hareketsiz görüntülere çerçeve denir. İki çerçeve arasında geçen süreye fps (frame per second) denir. Video oynatılırken ekranda saniyede kaç çerçeve gösterileceği fps ile belirlenir ve normal değer 24fps'dir. Video formatları ham ve sıkıştırılmış olarak ikiye ayrılır. AVI formatı sıkıştırılmamış, MPEG formatı ise sıkıştırılmış video formatına örnek olarak verilebilir.

Video dosyalarına veri gizleme işleminde video çerçevelerine ayrıştırılır. Her bir çerçeveye imgelere veri gizlemede kullanılan teknikler ile veri gizlenebilir. Videolar veri gizleme kapasitesi olarak en fazla yer sağlayan dosya formatlarıdır.

Bir videoya kayıt esnasında da veri gizlenebilir. Ünlü hazırladığı yüksek lisans tezinde web kamerası ile video kaydı sırasında gerçek zamanlı olarak veri gizlemeyi sağlayan bir sırörtme kütüphanesi tasarlamıştır [34]. Gerçek zamanlı videolara veri gizleme işleminin bir avantajı da videonun çekilme esnasında veri gizlendiği için örtü videonun başka birisinde olmamasından dolayı güvenlik arttırılmıştır. Orijinal

video dosyasının içinde gizli verinin barındırılmasından dolayı istatiksel sıraçma ataklarına karşı korunaklıdır [35].

2.3.4. Metin dosyalarında sırörtme

Metinlere gizleme işlemi metin içerisindeki boşluklardan tekrar eden metinlere kadar fazlalıklardan yararlanılarak gizleme tekniğidir. Shahreza bir çalışmasında Arapça ve Farsçadaki harflerin birçoğunda noktalama işaretlerinin kullanıldığını göstererek bu noktalama işaretlerin harf ile nokta arasındaki mesafeleri gizleme amacıyla kullanmıştır [36].

Spam e-postalar veri gizleme maksatlı kullanılabilir. Spammimic adlı internet sitesinde gizli veri spam e-postaya dönüştürülmektedir. Karşı taraf spam mesajı aldıktan sonra aynı internet sitesinde çözülecek gizli mesaja ulaşabilmektedir [37].

Metinlere gizleme işlemi günümüzde kullanıldığı gibi geçmişte de kullanılmıştır. İkinci Dünya Savaşında Alman bir casus tarafından yazılan telgrafda kullandığı bir veri gizleme örnek metni "Apparently neutrals protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretex for embargo on by-products, ejecting suets and vegetable oils." şeklindedir. Örnek metinde her kelimenin ikinci harfleri yan yana getirildiğinde "Pershing sails from NY June 1." mesajı ortaya çıkmaktadır [38].

2.4. Sırörtme İle İlgili Literatürdeki Çalışmalar

Sırörtmenin birçok uygulama alanı bulunmaktadır. İmge, ses ve video ortamlarına veri gizlenebilmektedir. Bu tez çalışmasında imgeler için veri gizleme yöntemleri çalışılmıştır.

Dereli, "Dilbilimsel Stegonanografi Yöntemleri Üzerine Bir Araştırma" adlı yüksek lisans tezinde dilbilimsel sırörtme (linguistic steganography) üzerine detaylı bir

araştırma yapılmıştır. Kullanılan yöntemler tarihçeler ve gerekçeleri üzerinde durulmuştur. Winstein, Nicetext, Wayner ve Atallah algoritmaları dışında, yeni bir yaklaşım olarak kriptografi kullanımını da benimseyen Markov zinciri tabanlı bir algoritma da incelenmiştir. Markov zinciri ve DES şifreleme sistemi algoritmasının performansı ve güvenliğinin artırılması için iyileştirme çalışmaları yapılmıştır [39].

Olcay, "Durağan Resimlere En Önemsiz Bit Yöntemiyle Veri Gizleme" adlı yüksek lisans tezinde görüntü dosyaları içersine LSB yöntemi ile veri gizlemiş ve farklı sıraçma yöntemleri ile test ederek sağlamlığını kontrol etmiştir. Sıraçma metodlarının sonuçlarını karşılaştırmıştır [18].

Lashkari, sayısal resim sırörtmesinde kullanılan algoritmaları incelemiş ve değerlendirmiştir [40].

Ünlü, yayınladığı yüksek lisans tezinde ortamdan ve yöntemden bağımsız bir sırörtme kütüphanesi tasarlamıştır. Kütüphane yardımıyla imge, ses ve video ortamlarına veri gizlenebilmektedir. Ayrıca canlı web kamera görüntüsünün içerisine veri gizlenmekte böylece gizlenecek mesaj bilgisinin uzunluğuna göre video çekilebilmektedir. Geliştirilen kütüphanede LSB, LSD ve LSBM yöntemlerine göre veri gizleme yapılabilmektedir. Kütüphane, tasarım desenleri kullanılarak kodlanmıştır. Böylece kütüphaneye yeni yöntemler ve özellikler kolaylıkla eklenebilir [34].

Al-Karawi, "İkili Görüntü Kullanarak İmza Sırörtme Modellerinin Geliştirilmesi" adlı yüksek lisans tezinde ikili görüntülere dayalı bir imza sırörtme modeli sunmaktadır. Bu yöntemle, sırlı imza görüntüsü elde etmek üzere düşük düzeyli özellikleri değiştirmeye dayalı bir şekilde kapak görüntüsü içerisine gizli imza verisini saklanmaktadır [41].

Erkin ve Örencik'in yayınladıkları sırörtme kütüphanesinde, imge sırörtmesine yönelik farklı yöntemler ile veri gizleme işlevine olanak sağlamaktadır. Kütüphaneyi geliştirilecek uygulamalarda kullanmak mümkündür [42].

Podilchuck ve Zeng, "Digital resimlerde görsel modelli damgalama" adlı çalışmalarında LSB tekniğinin zayıf noktalarını iyileştirebilmek için yeni bir yöntem önermişlerdir [26].

Cheddad, imge sırörtmesinde kullanılan algoritmaları inceleyerek karşılarştırmalarına yer vermiştir [43].

Jhonson, sırörtme yöntemlerini incelemiş ve sınıflandırarak detaylı bir araştırma yapmıştır [44].

Jung ve arkadaşları yeni bir sırörtme tekniği geliştirmişlerdir. Yöntemleri, imge dosyalarında çalışmaktadır. Buna göre, veri gizlemeden önce sayısal görüntü dosyasının çözünürlük değeri arttırılmış ve yeni elde edilen görüntüde eklenen kısımlara veri gizlenmesi sağlanmıştır. Böylece, orijinal görüntü dosyasının verileri bozulmamaktadır. Fakat bu yöntemin zayıf yanları mevcuttur. Bunlardan birisi dosyanın boyutu büyümektedir. Orijinal görüntü elde edilirse, veri gizlenmiş dosyanın bozulduğu kolaylıkla fark edebilir. Ayrıca kalite ölçütleri ile değerlendirildiğinde düşük bir performans göstermektedir [45].

Chrysochos ve arkadaşları imge sırörtme çalışması önermişler ve yaptıkları çalışmada imge histogramına dayalı yöntem ile veri gizlemişlerdir. Ancak bu yöntemde PSNR kalite ölçümü düşük çıkmaktadır [46].

Huang ve arkadaşı yaptıkları çalışmada imge dosyalarına veri gizlenmesiyle ilgili çalışmışlardır. Histogram temelli çalışmada kalite ölçütleri göz önünde bulundurulmamıştır [47].

Akar ve Varol RGB ağırlık gizleme yöntemi önermişler ve gizleme kapasitesini arttırdıklarını göstermişlerdir. PSNR kalite ölçüt değeri başarılı olan yöntemin histogram kalite ölçütü başarısız olmaktadır [24].

Bhatnagar ve ark. görüntü dosyaları için yeni bir damgalama yöntemi önermişlerdir. Yöntemin en büyük eksikliği damga verisi elde edilirken mutlaka orijinal görüntü dosyasına ihtiyaç duymasıdır. Veri gizleme yöntemlerinde orijinal dosyanın gerekliliği istenmeyen bir durumdur [48].

Gurijala ve arkadaşları, yaptıkları çalışmada ses damgalama tekniğini kullanmışlardır. Yöntemde, konuşma sinyalleri ile çalışmışlardır. Buna göre sinyalleri damgalamak için önerdikleri yöntemde doğrusal öngörü katsayılarında değişiklik yapmışlardır. Damgalamayı keşfetmek için orijinal konuşma sinyaline ihtiyaç duyulmaktadır [49].

Sağıroğlu ve Tunçkanat gri seviye resimlere LSB modifikasyonu yöntemiyle 4.dereceden veri gizlemişlerdir [50].

Tseng ve Chang yaptıkları çalışmada sıkıştırılmış JPEG resimler içerisine dönüştürme tekniğiyle veri gizlemişlerdir [51]. Brisbane ve Safavi-Naini paylaşımlı renk paleti ile yüksek kapasiteli veri gizleme yöntemi önermişlerdir [52]. Lee ve Chen, önerdikleri yöntemde 4.dereceden LSB bit gizleme ile resim içerisine %50 kapasite artışı ile veri gizlemeyi başarmışlardır [53].

Anderson'ın çalışmasında önerdiği metot örtü imgenin son bitlerinde küçük kesirli değişimler yapmaktadır. Örtü imgedeki bu değişiklikler karmaşık bir model ortaya çıkarmaktadır. Sıraçma yöntemleri tarafından algılanamayacak şekilde karmaşa oluşturulmuştur [54].

Resim dosyalarında yapılan birçok çalışma bulunmasına rağmen ses dosyaları üzerinde daha az çalışma yapılmıştır. Fakat son yıllarda ses dosyalarına veri gizleme

uygulamaları artmaktadır. Bu uygulamalar veri gizleme teknikleri LSB modifikasyonu ile dönüştürme tekniğiyle, tekrarlanan komutların varlığından faydalanarak ve sıkıştırma esnasında ses içerisine veri gizleme gibi yöntemleri kullanılarak geliştirilmiştir.

Gopalan, LSB yöntemini ile ses dosyaları içerisine veri saklanması üzerine çalışma yapmış ve kokpit sesi gibi ses dosyalarının içerisine daha fazla veri saklanabileceğini tesbit etmişleridir [32].

Yargıçoğlu, "Düşük Veri Hızlarında Çalışan Konuşma Kodlayıcılarına Gürbüz Bilgi Saklama Ve Damgalama" adlı doktora tezinde düşük hızlarda çalışan konuşma kodlayıcılarına damgalama ve sırörtme ile veri gizlenmesi çalışması yapılmıştır [55].

Yavuz, "Müzikle Şifreleme-Veri Gizleme Sistemi Tasarımı ve Gerçeklenmesi" adlı yüksek lisans tezinde müzik dosyalarının notalarından yararlanarak LSB yöntemi ile veri gizlemiş ve AES şifreleme ile de güvenliğini arttırmıştır. Bilginin geri elde edilmesi aşamasında herhangi bir anahtara ihtiyaç yoktur. Çünkü anahtar bilgisi de gizlenmiştir [30].

Chang ve Moskowitz, ses dosyalarının içine veri gizlemek için yöntemler incelemişler ve en başarılı yöntemin LSB yöntemi olduğunu göstermişlerdir. Fakat güvenlik zaafiyetini de belirtmişlerdir [56].

Kratzer ve arkadaşları çalışmalarında, karşılıklı sesli görüşme yapılırken ses bilgilerinin son bitlerine veri gizleme işlemini gerçekleştirmişlerdir. Veri gizleme yaparken bilgiler şifreli olarak gizlenmiştir [57].

Xu ve arkadaşları da sıkıştırılmış video görüntülerine sırörtme algoritması önermişlerdir [58].

Hartung, hem ham video verilerine hem de sıkıştırılmış video verilerine veri gizlemiştir. Gizleme işlemi sırasında her bir sıkıştırılmış çerçeve için DCT katsayılarına damgayı gizlemiştir [59].

Swanson, videolar için çoklu ölçekli damgalama yöntemini önermiştir. Her çerçeveye zamansal dalgacık dönüşümü uygulanır. Gizli veri, yeni oluşan çerçevelere gizlendikten sonra ters dönüşüm uygulanarak gerçek görüntü çerçevesi elde edilir. Bu yöntemde geri çıkarma işlemi için orijinal video dosyasına ihtiyaç vardır [60].

Çetin ve Özcerit videolar üzerine yeni bir sırörtme yöntemi geliştirmişlerdir. Geliştirilen yöntemde yazarlar videodaki çerçevelerde sabit ve hareketsiz bölgelere veri gizlemeyi önermişlerdir [61].

Kalker çalışmasında Hartung'un yaptığı gibi hareketli görüntüleri çerçevelere ayırarak her bir çerçeveye ayrı mesaj gizlemiştir. Jordan çalışmasında sıkıştırılmış videonun hareket vektörlerine veri gizlemeyi önermiştir. Hareket vektörleri üzerinden gizli veri geri çıkarılır [62].

Yalman imgeler için histogram temelli veri gizleme yöntemi geliştirmiştir. Önerilen yöntemde, örtü imgeye ait histogramın alt ve üst değerleri tespit edilir. Bu sınır değerlerine göre veri gizleme işlemi gerçekleştirilmektedir [22].

Literatür çalışmalarından da görüldüğü gibi birçok çalışma imgeler üzerine yapılmıştır. Literatürde en çok dikkat edilen nokta örtü bilgisinde en az değişiklik yaparak veriyi gizleyebilmektir. Tez çalışmasında önerdiğimiz yöntemlerde özellikle örtü bilgisinde en az değişiklik yaparak veriyi gizleme işlemi üzerinde durulmuştur. Bir sonraki bölümde tez çalışmasında geliştirilen sırörtme yöntemleri detayları ile anlatılacaktır.

BÖLÜM 3. YENİ YAKLAŞIM ESASLI ALTI SIRÖRTME YÖNTEMİ VE GERÇEKLEŞTİRİLMELERİ

Veri gizleme çalışmalarında yaygın olarak kullanılan en önemsiz bite gizleme tekniklerinde en önemli zayıflığı, gizli verinin tespit edilmesinin kolay olmasıdır. Bunun yanında diğer bir zayıflığı ise örtü ortamda sıraçma yöntemleri tarafından rahatlıkla görülebilen istatiksel izler bırakmasıdır. Bu nedenle gizli verinin tespit edilmesini zorlaştırmak için eşleştirme yöntemleri geliştirilmiştir.

Geleneksel yer değiştirme yönteminde (LSB) gizlenecek verilerin bitleri parça parça dosyanın her bir baytının son bitine saklanır ve son bitler 1 yada 0 olarak değiştirilmektedir. Bit gizlemede iki durum söz konusu olduğu için dosyanın son bitleri %50 oranında değişime uğramaktadır. Örneğin 1000 bit veri gizlendiğinde ortalama olarak taşıyıcı dosyanın son bitlerinde 500 bitlik bir veri değişmiş olur.

Eşleştirme yönteminde ise, dosyanın son bitleri doğrudan değiştirilmez. Bunun yerine dosyadaki bitler korunarak gizlenmek istenen veri taşıyıcı dosyada var olan bitler ile temsil edilmeye çalışılır. Gizlenecek bit bilgisi, örtü imgedeki sıradaki en önemsiz bit ile aynı ise herhangi bir değişiklik yapılmaz. Eğer bitler aynı değilse o zaman biti gizleyebilmek için yöntemin önerdiği şekilde değişiklik yapılır. Bu şekilde veri gizlendiğinde dosyada en az değişiklik oluştuğu için gizleme işlemin ortaya çıkarılması ihtimali azalır.

Sırörtmede veri çıkarma aşamasında iki yöntem bulunmaktadır. Bunlar tersinir ve tersinir olmayan yöntemlerdir. Tersinir yöntemlerde orijinal örtü imgesine ihtiyaç duyulmadan sadece sırlı imgeyle veri çıkarma işlemi yapılabilir. Gizlenen verilerin koordinatları sırlı imgede bulunmaktadır. Tersinir olmayan yöntemlerde ise veri çıkarma aşamasında orijinal örtü imgesine ihtiyaç duyulmaktadır. Tez çalışmasında eşleştirme yaklaşımı baz alınarak geliştirilen, imgeler için yeni bir veri gizleme

yöntemi ve bu yöntemden türetilen beş adet yöntem daha sunulmuştur. Tez çalışmasında geliştirilen birinci (EM-1), ikinci (EM-2) ve üçüncü (EM-3) sırörtme yöntemleri, gizli veri çıkarma işleminde orijinal örtü imgesine ihtiyaç duymaktadır. Sırörtme işleminde geri çıkarma aşamasında orijinal örtü imgesine ihtiyaç duyulması istenmeyen bir durumdur. Bu nedenle geri çıkarma işlemi için orijinal örtü imgeye ihtiyaç duymayan yaklaşık eşleştirme tabanlı tersinir veri gizleme yöntemleri olan dördüncü (EM-4), beşinci (EM-5) ve altıncı (EM-6) sırörtme yöntemleri geliştirildi.

3.1. Tersinir Olmayan Yöntemler

Tersinir olmayan yöntemlerden EM-1, EM-2 ve EM-3 yöntemleri gizli veri çıkarma işleminde orijinal örtü imgesine ihtiyaç duymaktadır. Orijinal imgeye ihtiyaç duyulmasının nedeni eşleşmeme durumunda kullanılmayan blokların tespit edilememesidir. Geri çıkarma işlemi sırasında orijinal imge ile sırlı imge kıyaslanarak kullanılan ve kullanılmayan bloklar tespit edilerek gizli veri çıkarılır.

3.1.1. 4-bitlik eşleştirme alanı tabanlı tersinir olmayan sırörtme yöntemi (EM-1)

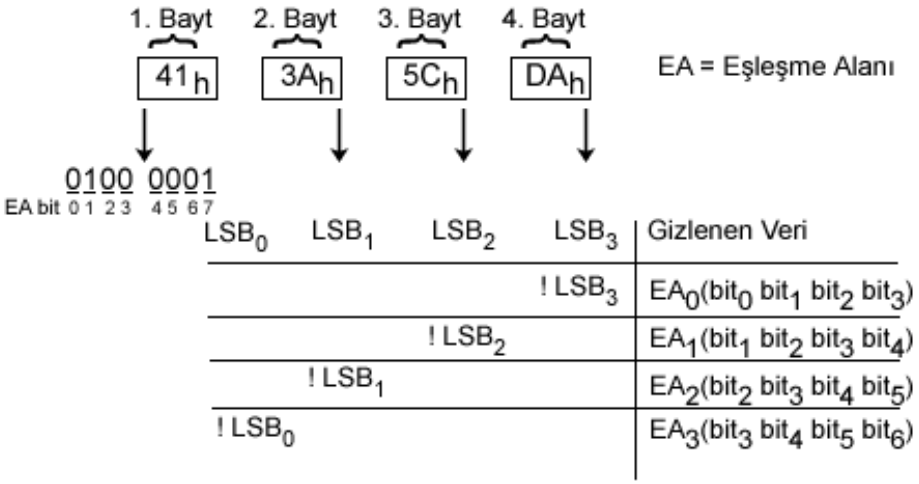
Tez çalışmasında geliştirilen 4-bitlik eşleştirme alanı tabanlı sırörtme yöntemi (EM-1) gizlenecek bitleri örtü imgedeki var olan bitler ile temsil ederek gizlemeye çalışır. Böylece geliştirilen yöntem örtü imgede en az değişiklik yaparak veri gizlenmiş olur.

EM-1 yönteminde örtü imge 4 baytlık bloklara bölünür. Bloktaki 1. bayt eşleştirme alanlarının oluşturulduğu referans baytı, diğer baytlar ise işaretleme işlemine destek vermek için kullanılan destek baytlarıdır. Bloktaki referans baytının soldan 7 biti eşleşme alanları için kullanılır. Şekil 3.1.'de eşleşme alanlarının oluşturulması gösterilmiştir. 7 bitlik bilgiden 4-bitlik 4 adet eşleşme alanı (EA) oluşur. Buna göre referans baytının soldan ilk 4 biti sırasıyla bit_0, bit_1, bit_2 ve bit_3 EA_1'i oluşturur. 1 bit sağa kaydırma (shift) yöntemi ile bit_1, bit_2, bit_3 ve bit_4 EA_2; bit_2, bit_3, bit_4 ve bit_5 EA_3 ve bit_3, bit_4, bit_5 ve bit_6 EA_4 olacak şekilde 4 farklı eşleşme alanı oluşur. Bu 4-bitlik

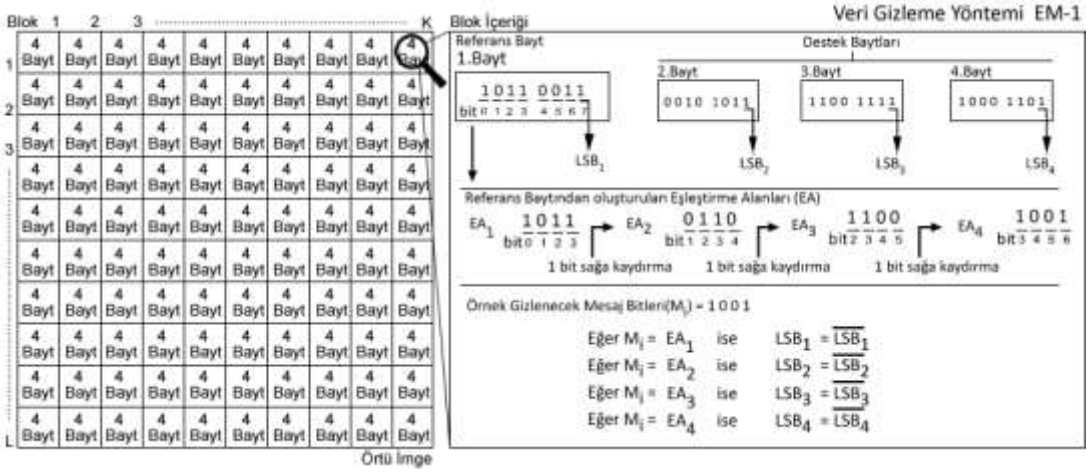
eşleşme alanlarına uygun olarak gizlenecek mesaj bilgisi de 4-bitlik gruplar halinde bölünür. Referans baytının en sağdaki biti olan bit_7 işaretleme amacıyla kullanılacağı için eşleştirme alanlarına dâhil edilmemiştir. Şekil 3.1.'de 4-baytlık bilginin LSB bitleri ile bir tablo oluşturulmuştur. Bu tabloya göre gizlenecek 4 bitlik mesaj parçası EA_0'a eşit ise gizli mesaj parçasının yerinin tespiti için LSB_3 terslenerek işaretleme yapılır. EA_1'e eşit ise LSB_2, EA_2'e eşit ise LSB_1, ve EA_3'e eşit ise LSB_0 terslenerek gizli bilginin konumu işaretlenir. Veri çıkarma aşamasında sırlı imgenin blokları orijinal örtü imgenin blokları ile karşılaştırılarak blok içindeki değişen LSB bitine göre gizlenen mesaj parçasının hangi eşleştirme alanında olduğu tespit edilecektir.

Şekil 3.2.'de ilk yöntem olan EM-1 yönteminin çalışma prensibi gösterilmiştir. Veri gizleme işleminde ilk olarak gizli mesajın soldan ilk 4 biti ile işleme başlanır ve eşleşme alanlarından herhangi birisine eşleşmesi kontrol edilir. Eğer eşleşme alanlarından hiçbirisine eşit değilse taşıyıcı dosyadaki bir sonraki 4 baytlık bloğa geçilir. Eşleşme alanlarından birisine eşitse Şekil 3.1.'deki gösterildiği gibi ilgili baytın son biti (LSB) terslenir. Örneğin EA_3'e eşit ise 3. baytın son biti terslenir. Böylece gizli mesaj bilgisini çıkarma işleminde kullanılmak için blokta saklanan bilginin hangi bitlerde olduğu işaretlenir. Bu yöntemle bir blokta 4-bitlik veri gizlendiğinde blokta 1 bitlik değişim oluşmaktadır.

Bloktaki referans baytının soldan 7 bitinden 4-bitlik 4 adet eşleşme alanı oluşmaktadır. 4 adet eşleşme alanından hangisiyle eşleştiği bilgisini işaretleyebilmek için 4 adet LSB bitine ihtiyaç vardır. Blok boyutu 5, 6, 7 ve 8-bayt olacak şekilde diğer durumlar da test edilmiş ve en verimli veri gizleme yönteminin 4-bayt olduğu tespit edilmiştir. Blok boyutu büyüdükçe oluşan blok sayısı azalmakta ve veri gizlenecek alan azalmaktadır. Ayrıca 4-bit eşleşme alanının eşleşme olasılığı $\frac{1}{16}$'dır. Blok boyutu büyüdükçe eşleşme olasılığı daha da azalmaktadır. Bu nedenlerden dolayı blok boyutu 4-bayt seçilmiştir.



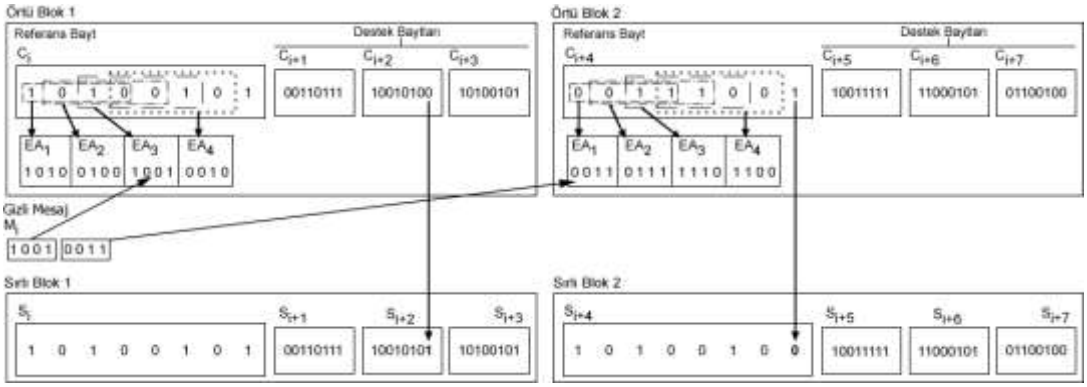
Şekil 3.1. EM-1 yönteminde eşleştirme alanlarının oluşturulması



Şekil 3.2. EM-1 yönteminin çalışma prensibi

Şekil 3.3.'de EM-1 yöntemi ile 1 baytlık veriyi gizleme işlemi örneklendirilerek gösterilmiştir. Şekil 3.3.'de verilen C_i örtü imgenin, S_i sırlı imgenin ve M_i'de gizlenecek mesajın *i*. baytını göstermektedir. Orjinal imgedeki her bloğun referans baytından yani C_i'.bayttan EA_1, EA_2, EA_3 ve EA_4 olmak üzere 4 adet eşleşme alanı oluşmaktadır. M_i'nin MSB'den başlayarak ilk 4 bitinin sıradaki bloğun eşleşme alanlarından birisine eşitliği kontrol edilir. Şekil 3.3.'de M_i'nin ilk 4 biti EA_3'e eşit olduğu için C_{i+2}'nin son biti terslenerek işaretlenir. Sırlı blokta S_{i+2}'nin son bitinin değiştiğini görebilirsiniz. Bir bloğa mesaj gizlendiğinde sadece bir bit değişiklik oluşur. Aynı şekilde M_i'nin son 4 biti ikinci bloktaki EA_1'e eşit olduğu için C_{i+4}'ün son biti terslenmiştir. Eğer gizlenecek 4-bitlik mesaj bilgisi sıradaki bloğun eşleşme

alanlarından hiç birisine eşit değilse bir sonraki bloğun eşleşme alanları kontrol edilir. Bu işlem tüm mesaj bitleri gizleninceye kadar devam eder.



Şekil 3.3. EM-1 yöntemiyle 1 bayt veri gizleme örneği

EM-1 veri gizleme yönteminin matematiksel ifadesi Eşitlik-3.1.'de verilmiştir.

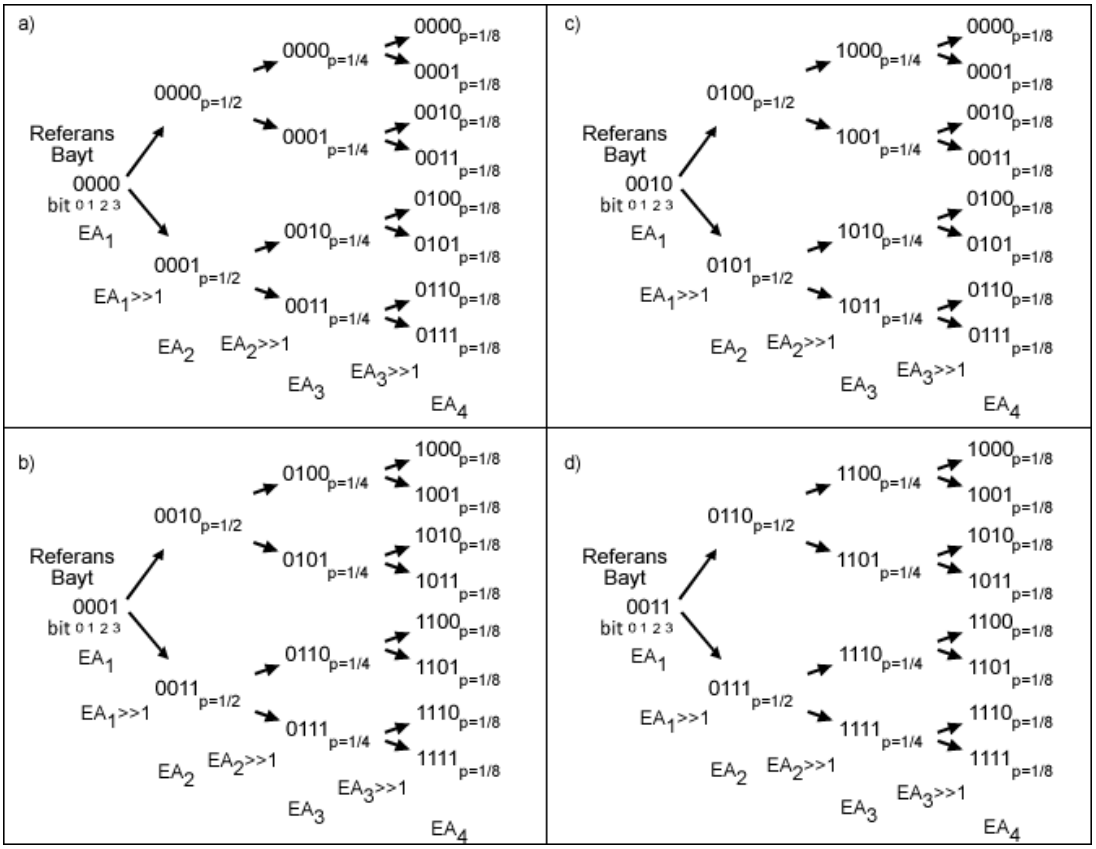
$$SB_i = \begin{cases} LSB(C_i) = Not\ LSB(C_i), & if\ M_i = EA_1 \\ LSB(C_{i+1}) = Not\ LSB(C_{i+1}), & if\ M_i = EA_2 \\ LSB(C_{i+2}) = Not\ LSB(C_{i+2}), & if\ M_i = EA_3 \\ LSB(C_{i+3}) = Not\ LSB(C_{i+3}), & if\ M_i = EA_4 \end{cases} \quad (3.1)$$

Eşitlik-3.1.'de SB sırlı bloğu, C_i örtü imgenin *i.* baytını, M_i gizlenecek 4-bitlik mesaj bitlerini, EA eşleşme alanlarını göstermektedir. Bununla beraber gizlenecek 4-bitlik mesaj bilgisi, çerçevedeki eşleşme alanlarından hiç birisine eşleşmeyebilir. Her bir eşleşme alanının gizlenecek mesaj bitleri ile eşleşmesinin olasılığı Eşitlik 3.2.'de verilmiştir.

$$p_{(EA_n)} = \frac{S(EA_n)}{S(E)} \quad (3.2)$$

4 bit ile en fazla 16 farklı durum oluşturabilir. 4-bitlik mesaj bilgisinin referans baytının (örn. C_i) ilk eşleşme alanına (EA_1) eşleşme olasılığı Eşitlik-3.2.'ye göre $p_{EA_1} = \frac{1}{16}$ olarak hesaplanmıştır.

Şekil 3.4.'te 0 ile 3 arası değerlere sahip eşleşme alanlarından bir bit sağa kaydırma yöntemiyle oluşabilecek eşleşme alanları gösterilmiştir. Şekil 3.4.a.'da gösterilen referans baytından üretilen EA_1 eşleşme alanı 0000 ise çerçeve bir bit sağa kaydırılarak elde edilen EA_2 eşleşme alanı, p= $\frac{1}{2}$ olasılıkla ya 0000 ya da 0001 olabilir. EA_2 eşleşme alanını 0000 olursa, bir bit sağa kaydırılarak elde edilen EA_3 eşleşme alanı ise p= $\frac{1}{4}$ olasılıkla ya 0000 ya da 0001 olabilir. Ya da EA_2 eşleşme alanının 0001 olursa, bir bit sağa kaydırılarak elde edilen EA_3 eşleşme alanı ise p= $\frac{1}{4}$ olasılıkla ya 0010 ya da 0011 olabilir. EA_3 eşleşme alanından bir bit sağa kaydırılarak üretilen EA_4 eşleşme alanı ise p= $\frac{1}{8}$ olasılıkla 0000-0111 arası 8 farklı değer alabilir. Şekil 3.b.' de EA_1'in 0001 olduğu durumda da EA_2 p= $\frac{1}{2}$ olasılıkla 0010 ya da 0011, EA_3 p= $\frac{1}{4}$ olasılıkla 0100, 0101, 0110 veya 0111, EA_4 ise p= $\frac{1}{8}$ olasılıkla 1000, 1001, 1010, 1011, 1100, 1101, 1110 veya 1111 değerlerini alabilir. Şekil 3. a, b, c ve d incelendiğinde EA_1 sırasıyla 0000 (0_{10}) – 0011 (3_{10}) arası değerleri alırken EA_2 0000 (0_{10}) – 0111 (7_{10}), EA_3 0000 (0_{10}) – 1111 (15_{10}) ve EA_4 0000 (0_{10}) – 1111 (15_{10}) değerlerini iki kere tekrar etmiştir. Bu bilgiler ışığında EA_1 0-15 arası değerler aldığında EA_2 0-15 arası değerleri 2 kez, EA_3 4 kez, EA_4 8 kez tekrar eder. Buna göre her eşleşme alanının olasılıkları Eşitlik-3.2.'e göre aşağıda hesaplanmıştır.



Şekil 3.4. Referans baytından üretilen eşleşme alanı ve bir bit sağa kaydırma yöntemiyle oluşabilecek eşleşme alanları örnekleri.

$$p_{EA_1} = \frac{1}{16} \qquad p_{EA_2} = \frac{2}{32} = \frac{1}{16} \qquad p_{EA_3} = \frac{4}{64} = \frac{1}{16} \qquad p_{EA_4} = \frac{8}{128} = \frac{1}{16}$$

Her eşleşme alanının eşleşme olasılığı birbiriyle aynı çıkmaktadır. Hesaplamalarda kullanılan bazı eşitlikler aşağıda verilmiştir.

Örtü İmge Boyutu(Bayt) = İmge Genişlik * Yükseklik * 3 (RGB) (3.3)

$$\text{Blok Sayısı} = \frac{\text{Örtü İ}mge\ Boyutu\ (Bayt)}{Blok\ Boyutu(Bayt)} \tag{3.4}$$

$$\text{Gizlenen Veri (Bayt)} = \frac{\text{Blok Sayı} * \text{Bir Blokta Gizlenen Bit Sayısı}}{8\ \text{bit}} \tag{3.5}$$

%Veri Gizlenen Bloğun Toplam Bloğa Oranı= $\frac{Veri\ Gizlenen\ Blok\ Sayısı}{Toplam\ Blok\ Sayısı}$x100 (3.6)

EM-1 yöntemi Matlab programında yazılarak uygulanmıştır. Eşleşme olasılığını istatiksel olarak inceleyebilmek için 500x400 boyutlarındaki bir renkli imgeye maksimum oranda EM-1 yöntemi ile başka bir renkli imge verisi gizlenmiştir. Gizleme sonucunda gizleme ilgili sayısal bilgiler yukarıda verilen eşitliklerle hesaplanmıştır. Örtü imge verisinin boyutu Eşitlik-3.3.'e göre 500*400*3=600.000 bayt olduğu için Eşitlik-3.4.'e göre $\frac{600.000}{4} = 150.000$ blok oluşur. Örtü imgesine 150.000 bloktan EM-1 yöntemiyle veri gizleme işlemi sonucunda 28.206 tane bloğa toplamda Eşitlik-3.5.'e göre $\frac{28.206*4\ bit}{8\ bit}$ = 14.103 bayt veri gizlenmiştir. Gizleme sonucunda gizlenecek 4-bitlik mesaj parçaları 7302 kez EA_1, 6193 kez EA_2, 7337 kez EA_3 ve 7374 kez EA_4 eşleşme alanıyla eşleşmiştir. 150.000 bloktan Eşitlik-3.6.'a göre her $\frac{28.206}{150.000}$ = %18,8 tanesine veri gizlenmiştir. Buna göre gizlenecek 4 bitli mesaj parçası tüm blokların %18,8'inin eşleşme alanıyla eşleşmiştir. Bu değerler örtü imgeye göre değişiklik göstermektedir.

Tez çalışmasında önerilen veri gizleme yöntemi EM-1'in veri gizleme sonucunda örtü imgede yaptığı değişim oranını hesaplayabilmek için kullanılan Eşitlik-3.7 aşağıda verilmiştir.

Değişim oranı $\epsilon = \frac{\text{Değişen Bit}}{\text{Gizlenen Bit}}$ (3.7)

EM-1 yönteminin Eşitlik-3.7.'ye göre değişim oranı $\epsilon = \frac{1}{4} = 0{,}25$ olur. Bu değer literatürdeki çalışmalara[1-5] göre başarılı bir değerdir.

EM-1 yönteminin veri gizleme kapasitesini hesaplayabilmek için kullanılan Eşitlik-3.8 aşağıda verilmiştir.

Gizleme Kapasitesi $\alpha = \frac{\text{Gizlenen Mesaj(Bayt)}}{\text{Örtü İmgesi (Bayt)}}$ (3.8)

EM-1 yönteminin Eşitlik-3.8.'e göre veri gizleme kapasitesi ortalama α=1/64 olur. Bu sonuca göre, EM-1 yöntemi bir imgeye boyutunun $\frac{1}{64}$ kadar veri gizleyebilmektedir. Örneğin 64KB boyuta sahip bir imgeye 1KB veri gizlenebilir. α değeri deneysel sonuçlarda kullanılan 150 imgenin EM-1 ile veri gizleme kapasitelerinin ortalamaları baz alınarak verilmiştir. Yöntem, karmaşık görüntüler içeren imgelerde çeşitliliğin fazla olması ve eşleşme olasılığının artması nedeniyle daha fazla oranda veri gizleyebilmektedir. EM-1 yöntemi, önerilen diğer yöntemlere göre örtü imgede en az değişiklik yaparken veri gizleme kapasitesi en düşüktür.

Aşağıda EM-1 yönteminin sözde kodu verilmiştir. Sözde kodda verilen kısaltmaların anlamları aşağıda verilmiştir.

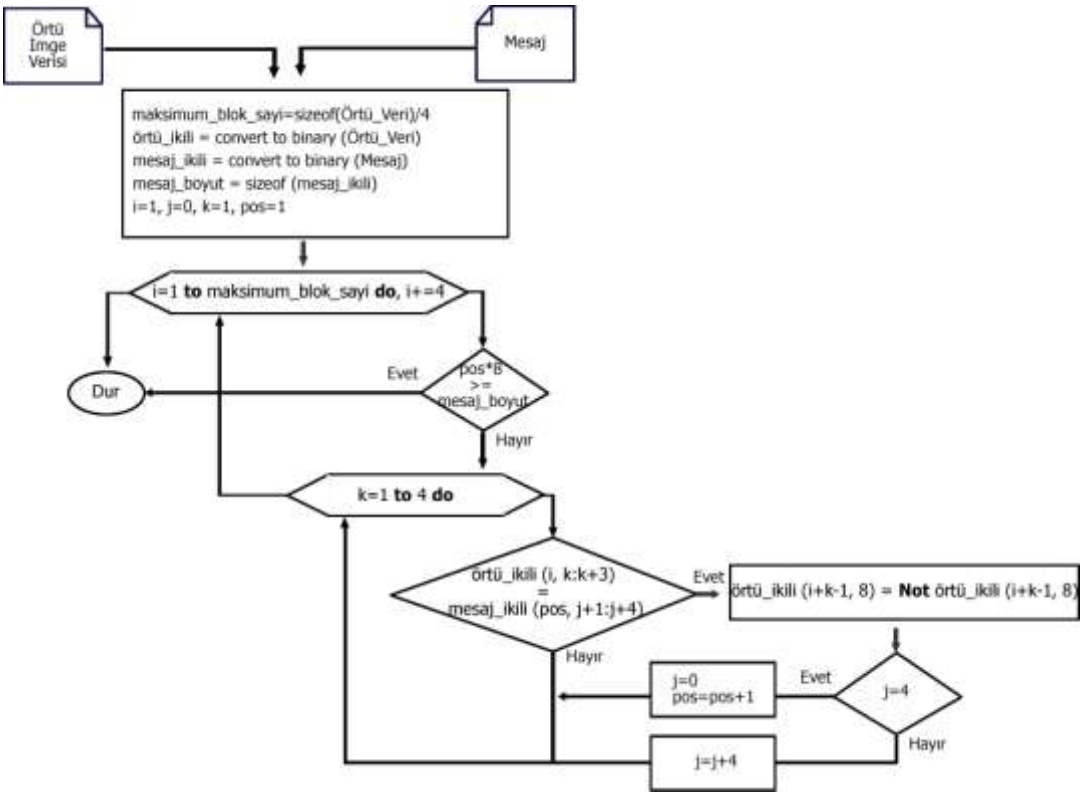
1. Örtü_Veri (i) : Örtü imgenin i. pozisyondaki baytın değerini
2. örtü_ikili (i, 1:4) : Örtü imgenin i. pozisyondaki baytın değerinin ilk 4 biti
3. örtü_ikili (i, 2:5) : Örtü imgenin i. pozisyondaki baytın değerinin ilk 4 bitinden itibaren 1 bit sağa kaydırma sonucu
4. örtü_ikili (i, 8) : Örtü imgenin i. pozisyondaki baytın değerinin en önemsiz biti (LSB)
5. örtü_ikili (i+1,8) : Örtü imgenin i+1. pozisyondaki baytın değerinin en önemsiz biti (LSB)
6. Mesaj (i): Gizlenecek mesaj verisinin i. pozisyondaki baytın değerini
7. mesaj_ikili (i,1:4) : Gizlenecek mesaj verisinin i. pozisyondaki baytın değerinin ilk 4 biti
8. mesaj_ikili (i,5:8) : Gizlenecek mesaj verisinin i. pozisyondaki baytın değerinin son 4 biti

EM-1 yöntemi sözde kodu;

```
1: fuction veri_gizleme_1 (Örtü_Veri , Mesaj)
2:        maksimum_blok_sayi = sizeof (Örtü_Veri) / 4
3:        örtü_ikili = convert to binary (Örtü_Veri)
4:        mesaj_ikili = convert to binary (Mesaj)
5:        mesaj_boyut = sizeof (Mesaj)
6:        i=1, j=0, k=1, pos=1
7:        while i <= maksimum_blok_sayi
8:              if  pos*8 >= mesaj_boyut then exit while
9:              for k=1 to 4 do
10:                   if  örtü_ikili (i, k:k+3) = mesaj_ikili (pos, j+1: j+4) then
11:                        örtü_ikili (i+k-1, 8) = Not örtü_ikili (i+k-1, 8)
12:                        if j==4 then j=0, pos=pos+1
13:                        else  j=j+4
14:                        end
15:                        exit for
16:                   end if
17:             end for
18:             i = i + 4
19:       end while
20: end function
```

EM-1 yönteminin sözde kodunda verilen veri_gizleme_1 fonksiyonu Örtü_Veri ve Mesaj adlı iki parametre alır. Bu parametreler (dosyalar) Matlab® tarafından onluk tabanda okunur. Daha sonra örtü verisi ve mesaj ikili sayı formuna dönüştürülür. Maksimum blok sayısı hesaplanır. Tüm blokları sırayla işleme alacak şekilde döngü kurulur. İlk 4-bitlik mesaj parçası alınarak işleme başlanır. Döngü işlemi sırasında gizlenecek i. pozisyondaki mesaj parçası örtü imgedeki i. pozisyondaki bloğun eşleşme alanları ile sırasıyla karşılaştırılır. i. pozisyondaki bloğun tüm eşlemle alanları için iç döngü (k) kurulur. İç döngü işlemi sırasında 10.satırdaki kod gizlenecek i. pozisyondaki mesaj parçası örtü imgedeki i. pozisyondaki bloğun k.

eşleşme alanı ile karşılaştırılır. Bu şekilde tüm eşleşme alanları ile sırasıyla karşılaştırılır. 10. satırdaki kod örtü verisinin k=1 iken i. pozisyonundaki baytının MSB'den ilk 2 bitini yani 1.eşleşme alanı EA_1'i gösterir. Benzer şekilde k=2 iken örtü verisinin i. pozisyonundaki baytının MSB'den ilk 2-3 arası 2.bitini yani 2.eşleşme alanı EA_2'i gösterir. Mesaj parçası hangi eşleşme alanıyla eşleşti ise ilgili baytın son biti terslenir. Eğer 1.eşleşme alanı olan EA_1 ile eşleşti ise k'nın aldığı değerlere göre i+k-1. pozisyondaki yani k=1 olduğu için i. pozisyondaki, EA_2 ile eşleşti ise i+1, EA_3 ile eşleşti ise i+2 ve EA_4 ile eşleşti ise i+3. pozisyondaki baytın son biti terslenerek çıkarma işlemi için işaretleme yapılır. Bu işlem mesajın tüm bitlerinin gizlenmesi veya tüm blokların bitmesiyle sonlanır.



Şekil 3.5. EM-1 yöntemi akış şeması

Şekil 3.5.'te EM-1 yönteminin akış şeması verilmiştir.

EM-1 yöntemiyle veri çıkarma işleminde aşağıdaki adımlar uygulanmaktadır.

1. Orijinal imge ve sırlı imge 4 baytlık bloklara ayrılır.
2. Orijinal imge ve sırlı imgedeki sıradaki bloktaki baytların son bitleri arasında fark olup olmadığı kontrol edilir.
3. Fark var ise hangi baytta değişim var ise referans baytının o sıradaki eşleştirme alanı gizli veri parçası olarak kaydedilir ve bir sonraki bloğa geçilir.
4. Fark yok ise bir sonraki bloğa geçilir.
5. Tüm bloklar bitene kadar bu işlemler devam eder.

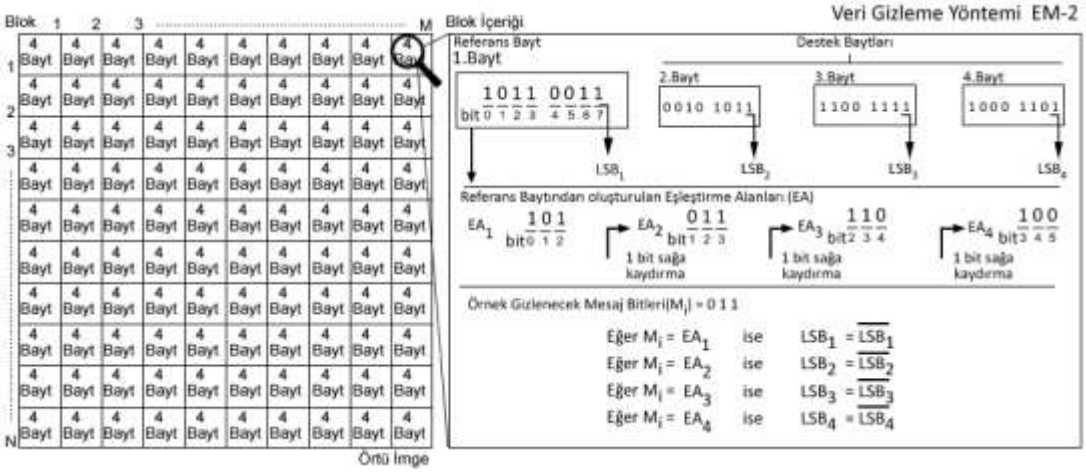
Elde edilen mesaj parçaları birleştirilerek gizli mesaj elde edilir. Yöntemin en büyük zayıflığı gizli veri çıkarılırken orijinal imgeye ihtiyaç duymasıdır.

3.1.2. 3-bitlik eşleştirme alanı tabanlı tersinir olmayan sırörtme yöntemi (EM-2)

Tez çalışmasında geliştirilen 4-bitlik eşleştirme alanı tabanlı sırörtme yöntemi (EM-1) yöntemi baz alınarak 3 bitlik eşleştirme alanı tabanlı sırörtme yöntemi (EM-2) türetilmiştir. Şekil 3.6.'da geliştirilen gizleme yöntemi EM-2'nin çalışma prensibi gösterilmiştir. EM-2 yönteminde taşıyıcı dosya 4 baytlık bloklara bölünür. EM-1 yönteminde olduğu gibi 1.baytın baştan (MSB) 7 biti eşleşme alanları için kullanılır. 7 bitlik bilgiden 3-bitlik 5 adet eşleşme alanı oluşmaktadır. Referans baytı ve 4 destek baytıyla birlikte 5 baytlık blok seçildiğinde veri gizleme performansının 4 baytlık bloğa göre daha düşük olduğu ölçülerek tespit edilmiştir. Bunun en büyük nedeni örtü verisi 5 baytlık bloklara bölündüğünde 4 bayta göre daha az blok sayısının oluşmasıdır. Böylelikle 5 baytlık blok sisteminde daha az blokla veri gizleme işlemi yapıldığından veri gizleme performansı 4 baytlık sistemden daha düşük olmaktadır. Bu nedenle 5 baytlık blok yerine 4 baytlık blok tercih edilmiştir. Buna göre referans baytı ve 3 destek baytıyla birlikte 4 baytlık blok seçilmiştir. Referans baytının ilk 3 biti bit_0, bit_1, bit_2 EA_1'i oluşturur. 1 bit sağa kaydırma (shift) yöntemi ile bit_1, bit_2, bit_3 EA_2; bit_2, bit_3, bit_4 EA_3 ve bit_3, bit_4, bit_5 EA_4 olacak şekilde 4 farklı eşleşme alanı oluşur. Referans baytından bit_6 ve bit_7 kullanılmamıştır.

Bunlardan bit_6 4 baytlık blok seçildiği için kullanılmazken bit_7 ise işaretleme amacıyla kullanılmaktadır.

3 bitlik eşleşme alanlarına uygun olarak gizlenecek mesaj bilgisi de 3-bitlik gruplar halinde bölünür. Mesajın ilk 3 biti ile işleme başlanır ve eşleşme alanlarından herhangi birisine eşleşmesi kontrol edilir. Eğer eşleşme alanlarından hiçbirisine eşit değilse örtü imgedeki bir sonraki 5 baytlık bloğa geçilir. Eşleşme alanlarından birisine eşitse Şekil 3.6.'daki gösterildiği gibi ilgili baytın son biti (LSB) terslenir. EM-1 yönteminden farkı bloklar için 5 bayt eşleşme alanları için ise 3-bit kullanılmasıdır. 3-bitlik mesaj bilgisinin eşleşme olasılığı EM-1 yönteminin kullandığı 4-bitlik mesaj bilgisinin eşleşme olasılığından daha yüksektir.



Şekil 3.6. EM-2 yönteminin çalışma prensibi

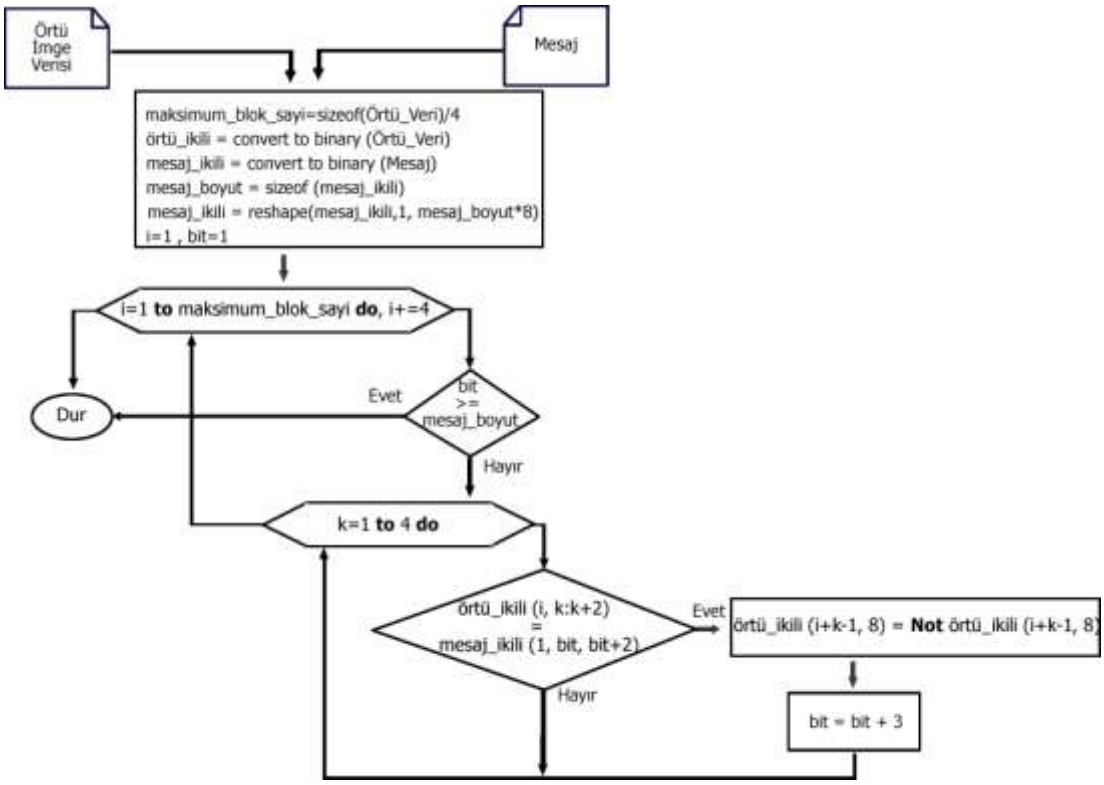
3 bit ile en fazla 8 farklı sayı yazılabilir. Buna göre EM-2 yönteminin eşleşme alanlarının olasılıkları aşağıda verilmiştir.

$$p_{MA_1} = \frac{1}{8},\ p_{MA_2} = \frac{2}{16} = \frac{1}{8},\ p_{MA_3} = \frac{4}{32} = \frac{1}{8},\ p_{MA_4} = \frac{8}{64} = \frac{1}{8},\ p_{MA_5} = \frac{16}{128} = \frac{1}{8}$$

EM-2 yöntemi Matlab programında yazılarak uygulanmıştır. Eşleşme olasılığını istatiksel olarak inceleyebilmek için 500x400 boyutlarındaki bir renkli imgeye maksimum oranda EM-2 yöntemi ile başka bir renkli imge verisi gizlenmiştir. İşlem

sonucunda gizleme ilgili sayısal bilgiler verilen eşitliklerle hesaplanmıştır. Her eşleşme alanının eşleşme olasılığı birbiriyle aynı çıkmaktadır.

Örtü imgenin boyutu Eşitlik-3.3.'e göre 500*400*3=600.000 bayt olduğu için Eşitlik-3.4.'e göre $\frac{600.000}{4} = 150.000$ blok oluşur. 150.000 bloktan 31.606 tane bloğa veri gizlenerek toplamda Eşitlik-3.5.'e göre $\frac{31.606*3\text{ bit}}{8\text{ bit}}$ =11.852 bayt veri gizlenmiştir. İşlem sonucunda gizlenecek 3 bitlik mesaj parçaları 3539 kez EA_1, 4264 kez EA_2, 7233 kez EA_3 ve 9032 kez EA_4 ve 7568 kez EA_5 eşleşme alanıyla eşleşmiştir. 120.000 bloktan Eşitlik-3.6.'a göre her $\frac{31.606*100}{150.000}$ = %21,07 tanesine veri gizlenmiştir. Buna göre, gizlenecek 3 bitli mesaj parçası tüm blokların 21,07'sinin eşleşme alanıyla eşleşmiştir.



Şekil 3.7. EM-2 yöntemi akış şeması

EM-2 yöntemiyle 3-bitlik veri gizlendiğinde örtü imgede 1-bitlik değişim oluşmaktadır. EM-2'nin değişim oranı $\epsilon = \frac{1}{3} = 0.33$, veri gizleme kapasitesi $\alpha = \frac{1}{48}$

bulunur. Bu sonuca göre, EM-2 yöntemi bir imgeye boyutunun $\frac{1}{48}$'i kadar veri gizleyebilmektedir. Örneğin, 48KB boyuta sahip bir imgeye 1KB veri gizleyebilir. α değeri deneysel sonuçlarda kullanılan 150 imgenin EM-2 ile veri gizleme kapasitelerinin ortalamaları baz alınarak verilmiştir. Yöntem, EM-1'de olduğu gibi karmaşık görüntüler içeren imgelerde çeşitliliğin fazla olması ve eşleşme olasılığının artması nedeniyle daha fazla oranda veri gizleyebilmektedir. EM-1 yöntemi, örtü imgede EM-2'ye yöntemine göre daha az değişiklik yaparken; EM-2 yöntemi de EM-1 yöntemine göre daha fazla veri gizleyebilir.

Şekil 3.7.'de EM-2 yönteminin akış şeması verilmiş ve ayrıca EM-2 yönteminin sözde kodu da verişmiştir. EM-2 yönteminin sözde kodunda verilen veri_gizle_2 fonksiyonu Örtü_Veri ve Mesaj adlı iki parametre alır. Bu parametreler (dosya) onluk olarak okunur. Örtü verisi ve Mesaj ikili (binary) sayı sistemli yapıya dönüştürülür. Maksimum blok sayısı hesaplanır. Tüm blokları sırayla işleme alacak şekilde dış döngü (i) kurulur. İlk 3-bitlik mesaj parçası alınarak işleme başlanır. Döngü işlemi sırasında gizlenecek i. pozisyondaki mesaj parçası örtü imgedeki i. pozisyondaki bloğun eşleşme alanları ile sırasıyla karşılaştırılır. i. pozisyondaki bloğun tüm eşlemle alanları için iç döngü(k) kurulur. İç döngü işlemi sırasında 12.satırdaki kod gizlenecek i. pozisyondaki mesaj parçası örtü imgedeki i. pozisyondaki bloğun k. eşleşme alanı ile karşılaştırılır. Bu şekilde tüm eşleşme alanları ile sırasıyla karşılaştırılır. 12. satırdaki kod, örtü verisinin k=1 iken i. pozisyonundaki baytının MSB'den ilk 2 bitini yani 1.eşleşme alanı EA_1'i gösterir. Benzer şekilde, k=2 iken örtü verisinin i. pozisyonundaki baytının MSB'den ilk 2-3 arası 2 bitini yani 2.eşleşme alanı EA_2'i gösterir. Mesaj parçası hangi eşleşme alanıyla eşleşti ise ilgili baytın son biti terslenir. Eğer 1.eşleşme alanı olan EA_1 ile eşleşti ise k'nın aldığı değerlere göre i+k-1. pozisyondaki yani k=1 olduğu için i. pozisyondaki, EA_2 ile eşleşti ise i+1, EA_3 ile eşleşti ise i+2 ve EA_4 ile eşleşti ise i+3.pozisyondaki baytın son biti terslenerek çıkarma işlemi için işaretleme yapılır. Bu işlem mesajın tüm bitlerinin gizlenmesi veya tüm blokların bitmesiyle sonlanır.

EM-2 yöntemi sözde kodu

```
1: fuction veri_gizleme_2 (Örtü_Veri , Mesaj)
2:        maksimum_blok_boyut = sizeof (Örtü_Veri) / 4
3:        örtü_ikili = binary (Örtü_Veri)
4:        mesaj_ikili =  convert_binary(Mesaj)
4:        mesaj_boyut = sizeof (mesaj_ikili)
5:        mesaj_ikili = reshape(mesaj_ikili,1, mesaj_boyut*8)
6:        i=1, bit=1
7:        while i <= maksimum_blok_boyut
8:              if  bit >= mesaj_boyut then
9:                    exit while
10:             end if
11:             for k=1 to 4 do
12:                   if  örtü_ikili (i, k:k+2) = mesaj_ikili (1, bit, bit+2) then
13:                         örtü_ikili (i+k-1, 8) = Not örtü_ikili (i+k-1, 8)
14:                         bit = bit + 3
15:                         exit for
16:                   end if
17:             end for
18:             i = i + 4
19:       end while
20: end function
```

EM-2 yöntemiyle veri çıkarma işleminde aşağıdaki adımlar uygulanmaktadır.

1. Orijinal imge ve sırlı imge 4 baytlık bloklara ayrılır.
2. Orijinal imge ve sırlı imgedeki sıradaki bloktaki baytların son bitleri arasında fark olup olmadığı kontrol edilir.

3. Fark var ise hangi baytta değişim var ise referans baytının o sıradaki eşleştirme alanı gizli veri parçası olarak kaydedilir ve bir sonraki bloğa geçilir.
4. Fark yok ise bir sonraki bloğa geçilir.
5. Tüm bloklar bitene kadar bu işlemler devam eder.

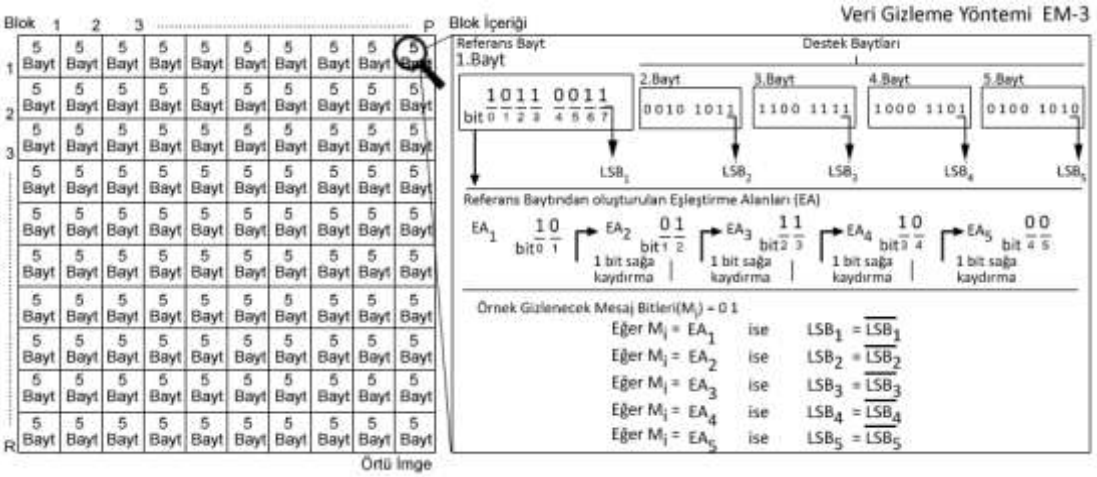
Elde edilen mesaj parçaları birleştirilerek gizli mesaj elde edilir.

3.1.3. 2-bitlik eşleştirme alanı tabanlı tersinir olmayan sırörtme yöntemi (EM-3)

Tez çalışmasında 2-bitlik eşleştirme alanı tabanlı sırörtme yöntemi (EM-3), 4-bitlik eşleştirme alanı tabanlı sırörtme yöntemi (EM-1) baz alınarak geliştirilmiştir. Şekil 3.8.'de geliştirilen gizleme yöntemi EM-3'ün çalışma prensibi gösterilmiştir. EM-3 yönteminde taşıyıcı dosya 5 baytlık bloklara bölünür. EM-1 yönteminde olduğu gibi 1.baytın baştan (MSB) 7 biti eşleşme alanları için kullanılır. 7 bitlik bilgiden 2-bitlik 6 adet eşleşme alanı oluşmaktadır. Referans baytı ve 5 destek baytıyla birlikte 6 baytlık blok seçildiğinde veri gizleme performansının 5 baytlık bloğa göre daha düşük olduğu ölçülerek tespit edilmiştir. Bunun en büyük nedeni, örtü verisi 6 baytlık bloklara bölündüğünde 5 bayta göre daha az blok sayısının oluşmasıdır. Böylelikle, 6 baytlık blok sisteminde daha az blokla veri gizleme işlemi yapıldığından veri gizleme performansı 5 baytlık sistemden daha düşük olmaktadır. Bu nedenle, 6 baytlık blok yerine 5 baytlık blok tercih edilmiştir. Buna göre, referans baytının ilk 2 biti bit_0 ve bit_1 EA_1'i oluşturur. 1 bit sağa kaydırma (shift) yöntemi ile bit_1, bit_2 EA_2; bit_2, bit_3 EA_3; bit_3, bit_4 EA_4 ve bit_4, bit_5 EA_5 olacak şekilde 5 farklı eşleşme alanı oluşur. Bu 2-bitlik eşleşme alanlarına uygun olarak gizlenecek mesaj bilgisi de 2-bitlik gruplar halinde bölünür. Mesajın ilk iki biti ile işleme başlanır ve eşleşme alanlarından herhangi birisine eşleşmesi kontrol edilir. Eğer eşleşme alanlarından hiçbirisine eşit değilse örtü imgedeki bir sonraki 5 baytlık bloğa geçilir. Eşleşme alanlarından birisine eşitse Şekil 3.8.'deki gösterildiği gibi ilgili baytın son biti (LSB) terslenerek çıkarma işlemi için işaretleme yapılır. EM-2 yönteminden farkı, eşleşme alanları için 2-bit kullanılmasıdır. 2-bitlik mesaj bilgisinin eşleşme

olasılığı, EM-2 yönteminin kullandığı 3 bitlik mesaj bilgisinin eşleşme olasılığından daha yüksektir. Şekil 3.8.'deki eşleştirme alanları incelendiğinde EA_1 ile EA_4'ün içeriği aynıdır. Bu örnekte olduğu gibi eşleştirme alanları aynı içeriğe sahip olabilir. Bu durum EM-3 yönteminin eşleşme olasılığını azaltmaktadır. Aynı durum EM-1 ve EM-2 yöntemleri içinde geçerlidir.2-bit ile en fazla 4 farklı sayı yazılabilir. Buna göre EM-3 yönteminin eşleşme alanlarının olasılıkları aşağıda verilmiştir.

$$p_{MA_1} = \frac{1}{4} \quad p_{MA_2} = \frac{2}{8} = \frac{1}{4} \quad p_{MA_3} = \frac{4}{16} = \frac{1}{4} \quad p_{MA_4} = \frac{8}{32} = \frac{1}{4} \quad p_{MA_5} = \frac{16}{64} = \frac{1}{4}$$



Şekil 3.8. EM-3 yönteminin çalışma prensibi

EM-3 yöntemi Matlab programında yazılarak uygulanmıştır. Eşleşme olasılığını istatiksel olarak inceleyebilmek için 500x400 boyutlarındaki bir renkli imgeye maksimum oranda EM-3 yöntemi ile başka bir renkli imge verisi 6 baytlık hem de 5 baytlık bloklar kullanılarak gizlenmiştir. Gizleme sonucunda gizleme ilgili sayısal bilgiler tez içinde verilen eşitliklerle hesaplanmıştır.

6 baytlık blok sistemine göre Eşitlik-3.4'e göre $\frac{600.000}{6} = 100.000$ blok oluşur. 100.000 bloktan 74.668 tane bloğa veri gizlenerek toplamda Eşitlik-3.5'e göre $\frac{74.668*2 \text{ bit}}{8 \text{ bit}} = 18.667$ bayt veri gizlenmiştir. Gizleme sonucunda gizlenecek 2-bitlik mesaj parçaları 16.421 kez EA_1, 12.348 kez EA_2, 15.723 kez EA_3 ve 13.299 kez EA_4, 9.657 kez EA_5 ve 7.220 kez EA_6 eşleşme alanıyla eşleşmiştir. 100.000 bloktan Eşitlik-3.6'a

göre her $\frac{74.668*100}{100.000} = \%74{,}68$ tanesine veri gizlenmiştir. Buna göre gizlenecek 2 bitli mesaj parçası tüm blokların %74,68'inin eşleşme alanıyla eşleşmiştir.

5 baytlık blok sistemine göre Eşitlik-3.4.'e göre $\frac{600.000}{5} = 120.000$ blok oluşur. 120.000 bloktan 78.632 tane bloğa veri gizlenerek toplamda Eşitlik-3.5.'e göre $\frac{78.632*2\text{ bit}}{8\text{ bit}} = 19.658$ bayt veri gizlenmiştir. Gizleme sonucunda gizlenecek 2-bitlik mesaj parçaları 17.293 kez EA_1, 14.274 kez EA_2, 18.738 kez EA_3 ve 16.349 kez EA_4 ve 11.978 kez EA_5 eşleşme alanıyla eşleşmiştir. 120.000 bloktan Eşitlik-3.6.'a göre her $\frac{78.632*100}{120.000} = \%65{,}52$ tanesine veri gizlenmiştir. Buna göre gizlenecek 2 bitli mesaj parçası tüm blokların %65,52'sinin eşleşme alanıyla eşleşmiştir.

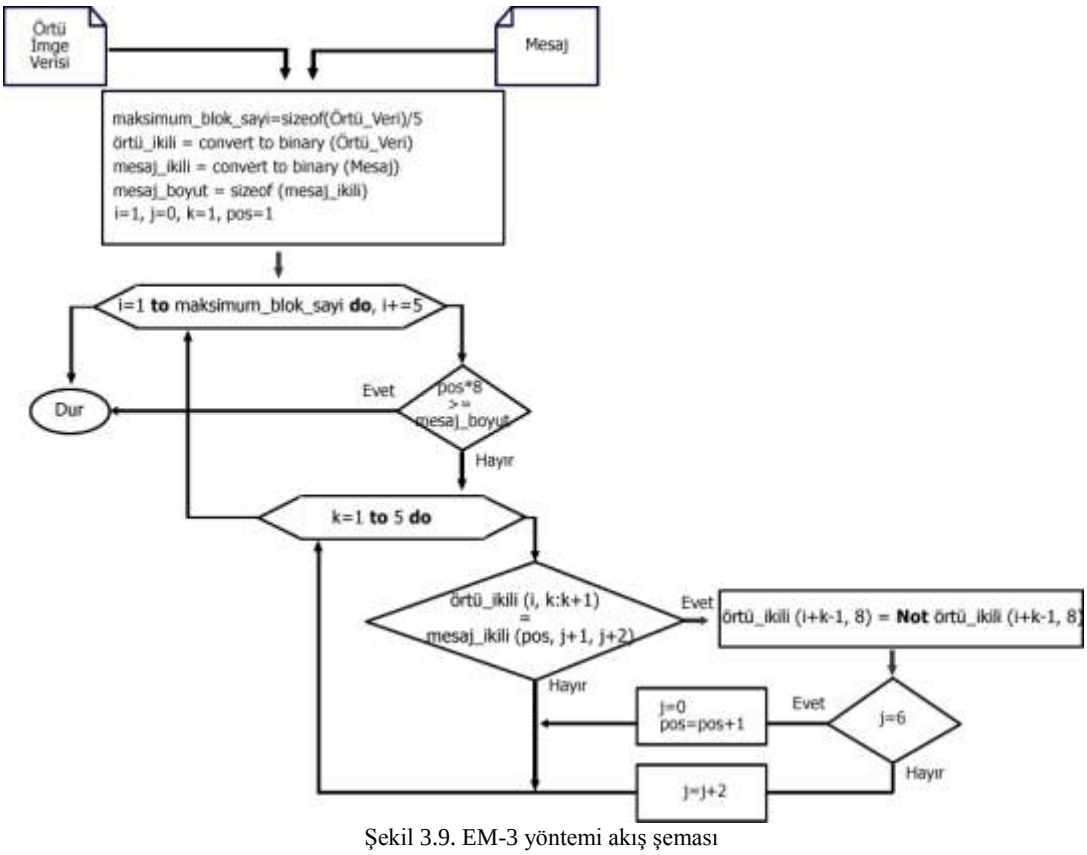
6 baytlık blok, 5 baytlık blok sistemine göre 991 bayt daha az veri gizlemiştir. Bu sonuçların gösterdiği üzere 5 baytlık sistem 6 baytlık sisteme göre daha fazla veri gizlemektedir. Bu nedenle, EM-3 yönteminde 5 baytlık sistem tercih edilmiştir. EM-3 yöntemiyle 2-bitlik veri gizlendiğinde 1 bitlik değişim oluşmaktadır. EM-3'nin değişim oranı $\epsilon = \frac{2}{1} = 2$, veri gizleme kapasitesi $\alpha = \frac{1}{37}$ bulunmuştur. Bu sonuca göre EM-3 yöntemi, bir imgeye boyutunun $\frac{1}{37}$'si kadar veri gizleyebilmektedir. Örneğin, 37KB boyuta sahip bir imgeye 1KB veri gizleyebilir. α değeri deneysel sonuçlarda kullanılan 150 imgenin EM-3 ile veri gizleme kapasitelerinin ortalamaları baz alınarak verilmiştir. Yöntem, EM-1'de olduğu gibi karmaşık görüntüler içeren imgelerde çeşitliliğin fazla olması ve eşleşme olasılığının artması nedeniyle daha fazla oranda veri gizleyebilmektedir. EM-2 yöntemi, örtü imgede EM-3'e yöntemine göre daha az değişiklik yaparken, EM-3 yöntemi de EM-2 yöntemine göre daha fazla veri gizleyebilir.

EM-3 yöntemi sözde kodu;

```
1: fuction veri_gizleme_3 (Örtü_Veri , Mesaj)
2:        maksimum_blok_sayi = sizeof (Örtü_Veri) / 5
3:        örtü_ikili = convert to binary (Örtü_Veri)
4:        mesaj_ikili =  convert_binary(Mesaj)
5:        mesaj_boyut = sizeof (mesaj_ikili)
6:        i=1, j=0, k=1, pos=1
7:        while i <= maksimum_blok_sayi
8:              if  pos*8 >= mesaj_boyut then
9:                 exit while
10:             end if
11:             for k=1 to 5 do
12:                   if  örtü_ikili (i, k:k+1) = mesaj_ikili (pos, j+1, j+2) then
13:                        örtü_ikili (i+k-1, 8) = Not örtü_ikili (i+k-1, 8)
14:                        if j==6 then j=0, pos=pos+1
15:                        else  j=j+2
16:                        end
17:                        exit for
18:                   end if
19:             end  for
20:             i = i + 5
21:       end while
22: end function
```

EM-3 yönteminin sözde kodunda verilen veri_gizleme_3 fonksiyonu Örtü_Veri ve Mesaj adlı parametreler Matlab tarafından okunan onluk formda dosya yapılarıdır. Örtü verisi ve Mesaj ikili sayı formuna dönüştürülür. Bu işlemi takiben, maksimum blok sayısı hesaplanır. Tüm blokları sırayla işleme alacak şekilde döngü kurulur. İlk 4-bitlik mesaj parçası alınarak işleme başlanır. Ana döngünün altına eşleşme alanı sayısı kadar iç döngü kurulur. Bu döngü tüm eşleşme alanlarını aramak için kurulmuştur. İç döngü işlemi sırasında 12.satırdaki kod gizlenecek mesaj parçası örtü

imgedeki i. pozisyondaki bloğun k. eşleşme alanı ile karşılaştırılır. Bu şekilde tüm eşleşme alanları ile sırasıyla karşılaştırılır. 12. satırdaki kod örtü verisinin k=1 iken i. pozisyonundaki baytının MSB'den ilk 2 bitini yani 1.eşleşme alanı EA_1'i gösterir. Benzer şekilde k=2 iken örtü verisinin i. pozisyonundaki baytının MSB'den ilk 2-3 arası 2 bitini yani 2.eşleşme alanı EA_2'i gösterir. Mesaj parçası hangi eşleşme alanıyla eşleşti ise ilgili baytın son biti terslenir. Eğer 1.eşleşme alanı olan EA_1 ile eşleşti ise k'nın aldığı değerlere göre i+k-1. pozisyondaki yani k=1 olduğu için i. pozisyondaki, EA_2 ile eşleşti ise i+1, EA_3 ile eşleşti ise i+2, EA_4 ile eşleşti ise i+3 ve EA_5 ile eşleşti ise i+4.pozisyondaki baytın son biti terslenerek çıkarma işlemi için işaretleme yapılır. Bu işlem mesajın tüm bitlerinin gizlenmesi veya tüm blokların bitmesiyle sonlanır. Şekil 3.9.'da EM-3 yönteminin akış şeması verilmiştir.



Şekil 3.9. EM-3 yöntemi akış şeması

EM-3 yöntemiyle veri çıkarma işleminde aşağıdaki adımlar uygulanmaktadır.

1. Orijinal imge ve sırlı imge 5 baytlık bloklara ayrılır.
2. Orijinal imge ve sırlı imgedeki sıradaki bloktaki baytların son bitleri arasında fark olup olmadığı kontrol edilir.
3. Fark var ise hangi baytta değişim var ise referans baytının o sıradaki eşleştirme alanı gizli veri parçası olarak kaydedilir ve bir sonraki bloğa geçilir.
4. Fark yok ise bir sonraki bloğa geçilir.
5. Tüm bloklar bitene kadar bu işlemler devam eder.

Elde edilen mesaj parçaları birleştirilerek gizli mesaj elde edilir.

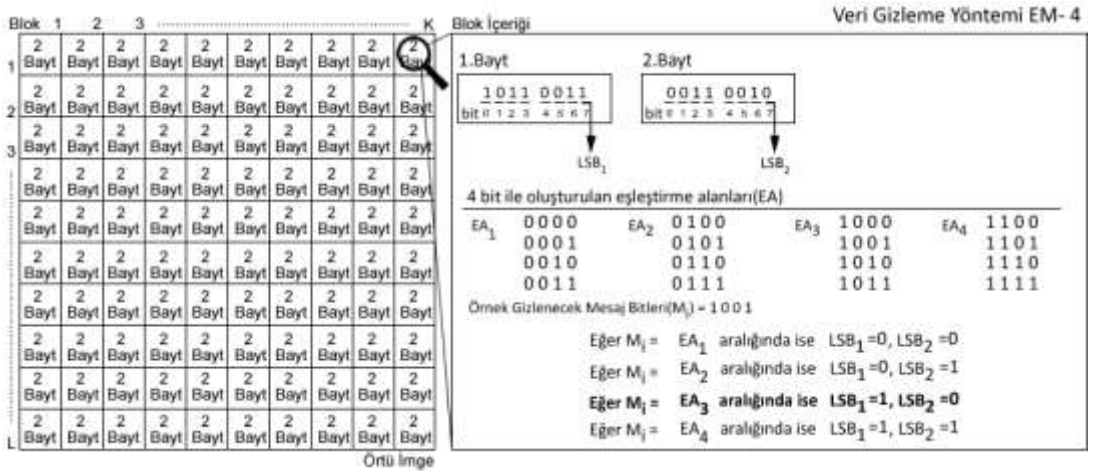
3.2. Tersinir Yöntemler

Tersinir yöntemlerde gizli imge orijinal imgeye gereksinim olmadan sırlı imgeden çıkartılılabilir. EM-4, EM-5 ve EM-6 yöntemleri tersinir yöntemlerdir. Bu yöntemlerde orijinal imgeye ihtiyaç olmadan sırlı imge içerisinden gizli imge çıkarılabilir. Sırlı imge içerisine gizlenen bilgiler çıkarma parametreleri ile birlikte gizlendiği için orijinal imgeye ihtiyaç duyulmaz. Oysa tersinir olmayan yöntemlerde ise gizli bilginin koordinatları sırlı imgeye gizlenmemiştir. Bu nedenle orijinal imgeye ihtiyaç duyulur.

3.2.1. 4-bitlik yaklaşık eşleştirme alanı tabanlı tersinir veri gizleme yöntemi (EM-4)

Önerilen yöntem 24-bit renkli imge içine kayıplı 24-bit renkli imge gizlemektedir. Önerilen yöntemin kapasitesi geleneksel LSB yöntemlerininin dört katıdır. Yöntem gizlenecek imgeyi 4-bitlik parçalar halinde 1 veya 2 yaklaşık değer ile geri elde edilecek şekilde gizler. Yani gizlenen imgenin piksel değerleri yaklaşık değerler ile geri elde edildiği için orijinal imgeye yakın renk tonları içeren gizli imge elde edilir.

Geliştirilen EM-4 yöntemi, örtü imgede gizlediği verinin %25 oranında bit değiştirmektedir. Bu değer EM-1 yöntemindeki değişim oranı ile aynıdır. Özet olarak 4 bit veriyi gizlerken örtü imgede 1 bit değişiklik oluşturmaktadır.



Şekil 3.10. EM-4 yöntemi çalışma prensibi

EM-4 yönteminde eşleştirme alanı 4-bitliktir. Gizli mesaj 4-bitlik parçalara bölünerek kodlanır. 4-bit ile onlu sistemde, 16 farklı sayı yazılabilir. 4-bitlik veri 2-bit ile temsil edilerek kodlanır. Buna göre 0 (0000_2) - 3 (0011_2) arası değerler 0 (00_2) ile; 4 (0100_2)-7 (0111_2) arası değerler 1 (01_2) ile; 8 (1000_2) - 11 (1011_2) arası değerler 2 (10_2) ile; ve 12 (1100_2) – 15 (1111_2) arası değerler ise 3 (11_2) ile kodlanır. EM-1 - EM-3 yöntemlerinde tam eşleşme arandığı için eşleşme olmadığı durumlarda kullanılmayan bloklar oluşmakta ve bu durum gizli veriyi çıkarma aşamasında problem oluşturmaktadır. Bu nedenle EM-1, EM-2 ve EM-3 yöntemi geri çıkarma işleminde orijinal imgeye ihtiyaç duymaktadır. EM-4 yönteminden gizlenecek veri ortalama değer ile eşleştirilerek temsil edilmekte ve böylece eşleşmeyen durum oluşmamaktadır. Gizli veri yaklaşık değerler ile temsil edildiği için geri elde edilen imgenin görüntü kalitesi olarak bozulmaya uğramaktadır. Fakat alıcı taraf gizli bilgiyi görüntü kalitesi düşük olsa bile anlayabilmektedir.

Şekil 3.10.'da EM-4 yönteminin çalışma prensibi gösterilmiştir. Veri gizleme işleminde örtü imge, 4-bitlik parçalara bölünerek gizlenir. Gizlenecek 4-bitlik veri dört eşleşme alanından hangisinin aralığına giriyorsa ona göre örtü imgede iki bitlik

işaretleme yapılır. Kesin eşleşme olduğu için, gizleme işlemi 2-baytlık bloklar halinde ilerler. Önceki yöntemlerde olduğu gibi eşleşmeme durumu yoktur. Şekil 3.10.'daki örnek gizlenecek mesaj bilgisi 1001_2 EA_3'ün aralığında olduğu için LSB_1 1 ile LSB_2 ise 0 ile değiştirilerek gizli veri işaretlenir. Böylece gizli mesaj bilgisini çıkarma işleminde kullanılmak için blokta saklanan veri işaretlenir. Bu yöntemle 4-bitlik veri gizlendiğinde 1 bitlik değişim oluşmaktadır. Normalde 2-bit değişiklik yapılmasına rağmen LSB ile gizleme yönteminde %50 ihtimalden dolayı 1 bit değişim olur.

EM-4 yöntemini matematiksel olarak ifade edecek olursak; bir imge 2-bayt boyutunda **n** adet bloğa bölünür ve her bloğun $\mathbf{SB_i, i=1\ldots n,}$ 2 adet LSB biti bulunur. Bloklardaki bu 2-bitlik veri işaretleme amacıyla kullanılarak her bloğa 4-bitlik mesaj parçası, $\mathbf{M_i}$ gizlenebilir. Buna göre bir imgeye **n*4** bit uzunluğunda mesaj gizlenebilir. Bunu bir örnekle gösterecek olursak; 600KB bir imgede 300.000 blok oluşur. Toplamda 300.000*4 1.200.000 bit veri gizlenebilir. 1.200.000/8 = 150.000 bayt/1024 = 146,5KB veri gizlenebilir.

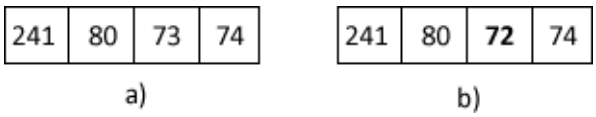
Eşitlik 3.9.'da sırlı blokta (SB) iki bitlik LSB biti ile 4-bitlik bilginin nasıl temsil edildiği verilmiştir. Gizlenecek 4-bitlik mesaj parçası hangi eşleştirme alanı aralığına giriyorsa ona göre **SB**'deki iki bitlik veri Eşitlik 3.9.'da gösterildiği gibi değiştirilir.

$$SB = \begin{cases} LSB(C_i) = 0, LSB(C_{i+1}) = 0 \;\; EA_1 = \{ M_i \mid 0 \leq M_i \leq 3, \;\; M_i \in Z\} \\ LSB(C_i) = 0, LSB(C_{i+1}) = 1 \;\; EA_2 = \{ M_i \mid 4 \leq M_i \leq 7, \;\; M_i \in Z\} \\ LSB(C_i) = 1, LSB(C_{i+1}) = 0 \;\; EA_3 = \{ M_i \mid 8 \leq M_i \leq 11, \;\; M_i \in Z\} \\ LSB(C_i) = 1, LSB(C_{i+1}) = 1 \;\; EA_4 = \{ M_i \mid 12 \leq M_i \leq 15, \;\; M_i \in Z\} \end{cases} \quad (3.9)$$

Eşitlik-3.9.'da **SB** sırlı bloğu, $\mathbf{C_i}$ örtü imgenin **i.** baytını, **LSB** $\mathbf{(C_i)}$ **i.**baytın en önemsiz bitini, **EA** eşleştirme alanlarını, $\mathbf{M_i}$ gizlenecek 4-bitlik mesaj bitinin onluk karşılığını göstermektedir. Eşitlik-3.9.'a 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.

Örnek olarak; 10010010_2 bilgisini EM-4 yöntemi ile gizlemek için gizli veri 4-bitlik parçalara bölünerek kodlanır. İlk 4-bitlik parça 1010_2 bilgisi onluk sistemde 10'a karşılık gelmektedir. 10 bilgisi EA_3'ün eşleştirme aralığına girdiği için işaret biti 10_2 olmaktadır. Şekil 3.11.'de örtü imgenin 2 bloğu gösterilmiştir. Buna göre, örtü imgedeki ilk bloktaki 241,80 piksel bilgilerinin son bitleri ile 10_2 değeri elde edilir. EM-4 yönteminde ilk 4-bitlik veri gizlendiğinde ve 10_2 işaretlemesi yapıldığında bloktaki herhangi bir değişiklik olmaz. İkinci 4-bitlik mesaj parçası, 0010_2 bilgisi onluk sistemde 2'ye karşılık gelmektedir ve EA_1'in eşleştirme aralığına girdiği için işaret biti 00_2 olmaktadır. Örtü imgenin ikinci bloğundaki 73, 74 piksel değerlerinin son bitleri ile 10_2 değeri elde edilir. Sıradaki 4-bitlik bilginin işaret biti 00_2 işaretlemesi yapıldığında bloktaki son durum 72, 74 olur. Örneğimizde 8-bit veri gizlenmesine rağmen sadece 1 bit veri değişikliği olmuştur.

EM-4'nin değişim oranı $Є = \frac{1}{4} = 0{,}25$, veri gizleme kapasitesi $\alpha = \frac{1}{4}$ bulunmuştur. Bu sonuca göre EM-4 yöntemi bir imgeye boyutunun %25'i kadar veri gizleyebilmektedir. Örneğin 60KB boyuta sahip bir imgeye 15KB'a kadar veri gizleyebilir. α değeri deneysel sonuçlarda kullanılan 150 imgenin EM-4 ile veri gizleme kapasitelerinin ortalamaları baz alınarak verilmiştir.



Şekil 3.11. EM-4 yöntemi ile veri gizleme a) Orijinal imgenin iki bloğu b) 8-bit gizlenmiş sırlı imgenin iki bloğu

Aşağıda EM-4 yönteminin sözde kodu verilmiştir. EM-4 yönteminin sözde kodunda verilen veri_gizleme_7 fonksiyonu Örtü_Veri ve Mesaj adlı iki parametre alır. Örtü verisi ve mesaj ikili sayı formuna dönüştürülür. Ardından maksimum blok sayısı hesaplanır ve tüm blokları sırayla işleme alacak şekilde döngü kurulur. İlk 4-bitlik mesaj parçası alınarak işleme başlanır. Gizlenecek 4-bitlik mesaj parçası onluk sayıya 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 ile değiştirilir. Bu şekilde

gizlenecek bitler iki bite indirgenerek işaretlenir. Ve 4'er bit 4'er bit gizlenecek mesaj, örtü imgeye gömülür.

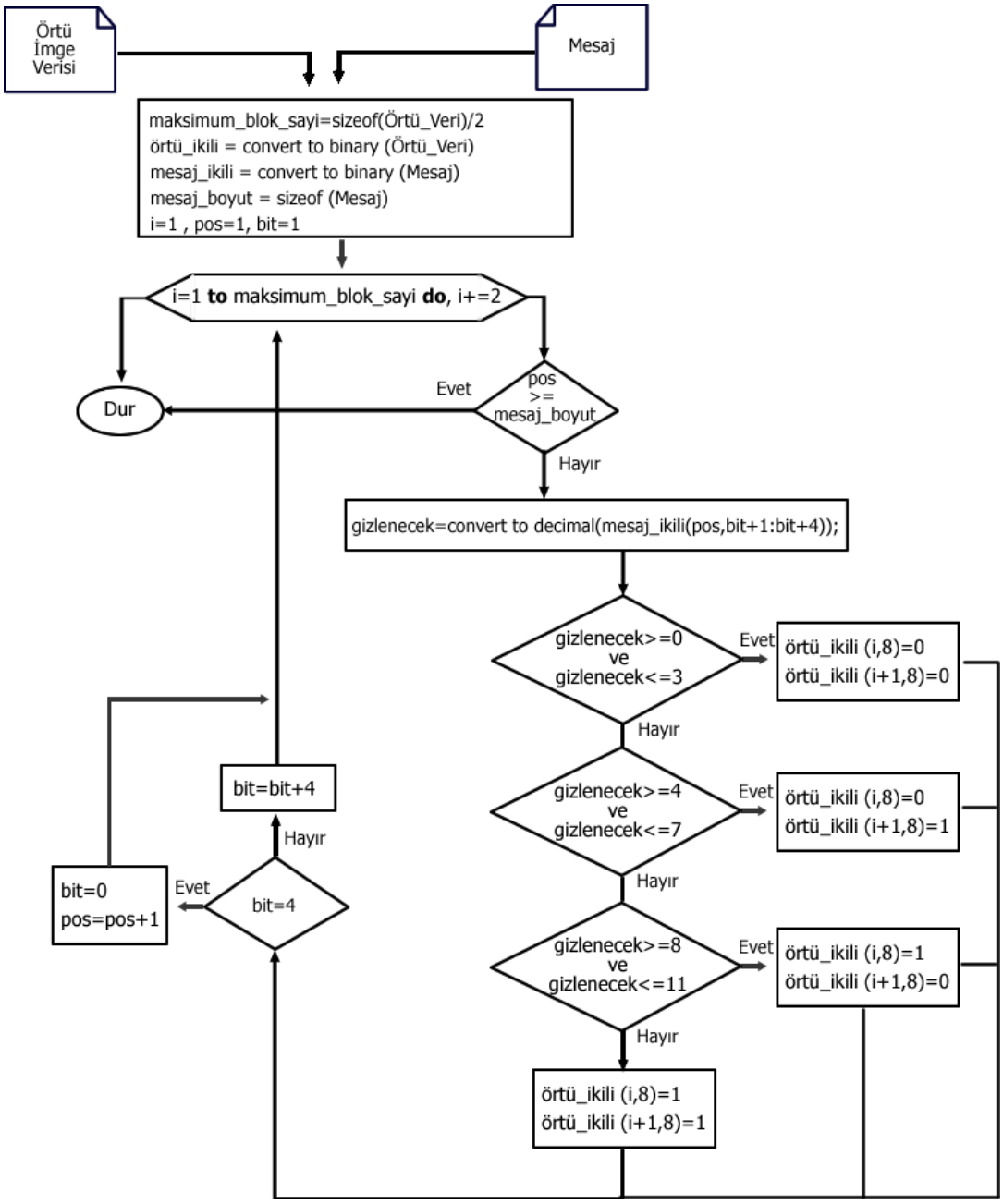
EM-4 yöntemi sözde kodu;

```
1: fuction veri_gizleme_4 (Örtü_Veri , Mesaj)
2:        maksimum_blok_sayi = sizeof (Örtü_Veri) / 2
3:        örtü_ikili = convert to binary (Örtü_Veri)
4:        mesaj_ikili = convert to binary (Mesaj)
5:        mesaj_boyut = sizeof (Mesaj)
6:        i=1, pos=1, bit=0
7:        while i <= maksimum_blok_sayi
8:              if  pos >= mesaj_boyut then exit while
9:              gizlenecek=convert to decimal(mesaj_ikili(pos,bit+1:bit+4));
10:             if  gizlenecek>=0 && gizlenecek<=3 then
11:               örtü_ikili (i, 8) = 0; örtü_ikili (i+1, 8) = 0;
12:             else if  gizlenecek>=4 && gizlenecek<=7 then
13:                     örtü_ikili (i, 8) = 0; örtü_ikili (i+1, 8) = 1;
14:                 else if  gizlenecek>=8 && gizlenecek<=11 then
15:                         örtü_ikili (i, 8) = 1; örtü_ikili (i+1, 8) = 0;
16:                     else
17:                         örtü_ikili (i, 8) = 1; örtü_ikili (i+1, 8) = 1;
18:                     end if
19:                 end if
20:             end if
21:             if  bit==4  then
22:                 bit=0; pos=pos+1;
23:             else
24:                bit=bit+4;
25:             end if
26:             i = i + 2
27:       end while
28: end function
```

Şekil 3.12.'de EM-4 veri gizleme yönteminin akış diyagramı verilmiştir. EM-4 yöntemiyle veri çıkarma işleminde aşağıdaki adımlar uygulanmaktadır.

1- Sırlı imge 2 baytlık bloklara ayrılır.
2- Sıradaki bloğun son bitleri okunarak onluk sayıya çevrilir.
3- Onluk sayı 0 ise gizli mesaj parçası 0010_2 elde edilir.
4- Onluk sayı 1 ise gizli mesaj parçası 0110_2 elde edilir.
5- Onluk sayı 2 ise gizli mesaj parçası 1010_2 elde edilir.
6- Onluk sayı 3 ise gizli mesaj parçası 1110_2 elde edilir.
7- Elde edilen mesaj parçaları birleştirilerek gizli mesaj elde edilir.

Sırlı imgenin ilk 20 biti gizli mesaj bilgisinin boyutunu göstermektedir. Gizli mesaj bilgisinin boyutu okunduktan sonra geri çıkarma işleminde de gizli mesajın boyutu kadar sırlı blok okunur.



Şekil 3.12. EM-4 veri gizleme yöntemi akış diyagramı

EM-4 yöntemi veriyi gizlerken yaklaşık eşleşme yöntemi ile gizlemektedir. Veri çıkarma işleminde bloktaki 2 baytlık bilginin son bitlerine gizlenen bilgiye göre gizli mesaj parçası yaklaşık olarak elde edilir. Aşağıda EM-4 yönteminin veriyi geri çıkarma işleminin sözde kodu verilmiştir.

EM-4 yönteminin veri geri çıkarma işlemi sözde kodu;

```
1: function veri_cikarma_4 (Sırlı_Veri)
2:         maksimum_blok_sayi = sizeof (Sırlı_Veri) / 2
3:         sırlı_ikili = convert to binary (Sırlı_Veri)
4:         mesaj_boyut = convert to decimal(LSB(20bit))
5:         i=1, pos=1, bit=0
6:         while i <= maksimum_blok_sayi
7:                 if  sizeof (gizli_mesaj) >= mesaj_boyut then exit while
8:                 gizli_veri=convert to decimal ([sırlı_ikili(i,8), sırlı_ikili(i+1,8)]);
9:                 if  gizli_veri ==0 then
10:                   mesaj_parcasi=0010;
11:                else if  gizli_veri ==1 then
12:                        mesaj_parcasi=0110;
13:                    else if  gizli_veri ==2 then
14:                            mesaj_parcasi=1010;
15:                        else
16:                            mesaj_parcasi=1110;
17:                        end if
18:                    end if
19:                end if
20:                mesaj=mesaj+mesaj_parcasi;
21:                if uzunluk (mesaj)==8 then
22:                   gizli_mesaj=gizli_mesaj+mesaj;
23:                end if
24:                i = i + 2
25:         end while
26: end function
```

Geri çıkarma işlemini bir örnek ile açıklanırsa; Şekil 3.13.'de gösterilen sırlı imgenin iki bloğuna gizlenen verinin geri elde edilmesi için ilk bloktaki 241,80 piksel bilgilerinin son bitleri ile 10_2 değeri elde edilir. 10_2 bilgisinin onluk karşılığı 2'nin

karşılığında elde edilen gizli mesaj parçası 1010_2 olur. İlk bloğa gizlenen veri ise 1001_2 idi. 1010_2'in onluk karşılığı 10 iken 1001_2'in onluk karşılığı 9'dur. Böylece 1 değer sapma ile gizli mesaj parçası elde edilmiştir. İkinci bloğun 72, 74 piksel bilgilerinin son bitleri ile 00_2 bilgisi elde edilir. 00_2 bilgisinin onluk karşılığı 0'ın karşılığında elde edilen gizli mesaj parçası 0010_2 olur. Böylece gizlenen mesaj parçası tam olarak geri elde edilir.

a) b)

Şekil 3.13. EM-4 yöntemiyle veri gizleme a) EM-4 yöntemiyle gizlenen imge b) EM-4 yöntemiyle geri çıkarılan

Şekil 3.13.a.'da EM-4 yöntemiyle gizlenmiş imgenin orijinal hali verilmiştir. Şekil 3.13.b.'de ise EM-4 yöntemiyle geri elde edilmiş imge verilmiştir. Geri elde edilen imgenin kalitesi, yaklaşık eşleştirme işleminden dolayı düşmüştür. Fakat karşı tarafa iletilen mesaj anlaşılmayacak kadar bozulmamıştır.

Geliştirilen yöntem ile mesaj iletilmek istenirse iletilecek mesaj Paint® gibi resim düzenleyici programlar ile hazırlanıp imge içerisine imge olarak gizlenebilir. Şekil 3.14.'de imge içerisine gizlenmiş bir mesajın imge olarak gizlenmeden önceki hali ve farklı paremetrelerle geri elde edilmiş halleri verilmiştir. Tablo 3,1.'de gizlenen bilginin geri elde edilmesinde aralık değerlerine göre gizli verinin değişimi verilmiştir. Düşük aralık değerinde gizlenen veri aralığındaki en düşük değer, orta aralık değerinde gizlenen veri aralığındaki orta değer ve yüksek aralık değerinde ise gizlenen veri aralığındaki en yüksek değer geri elde edilecek değer olarak seçilmiştir. Buna göre eğer düşük aralık seçilerek gizli imge elde edildiğinde gizli imgeye koyu

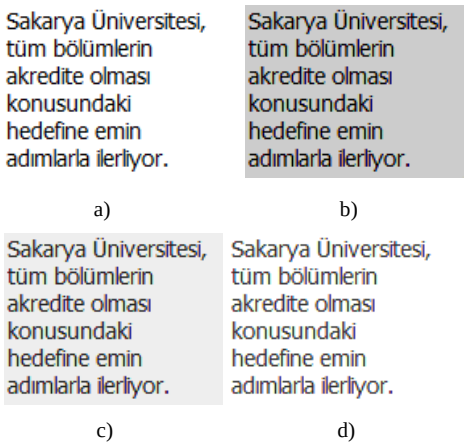
renk tonları hâkim olurken, orta aralık seçildiğinde gizli imge ortalama renk tonlarına ve yüksek aralık değeri seçilerek geri çıkarma işlemi yapıldığında gizli imge açık renk tonları hâkim olarak çıkarılmaktadır.

Tablo 3.1. Gizlenen değer ve geri elde değerin aralık değerlerine göre değişimi

Gizlenen değer	Geri elde edilen değer	Aralık
0-3	0	Düşük
0-3	2	Orta
0-3	3	Yüksek
4-7	4	Düşük
4-7	6	Orta
4-7	7	Yüksek
8-11	8	Düşük
8-11	10	Orta
8-11	11	Yüksek
12-15	12	Düşük
12-15	14	Orta
12-15	15	Yüksek

Şekil 3.14.'de beyaz arka fonunda siyah renk yazı fontu mesaj içeren imgenin gizlenmeden önceki ve farklı aralık değerleri seçilerek geri elde edilen halleri verilmiştir. Şekil 3.14.a.'da orijinal gizlenen imge, Şekil 3.14.b.'de düşük aralık seçilerek, Şekil 3.14.c.'de orta aralık seçilerek ve Şekil 3.14.d.'de yüksek aralık seçilerek çıkarılan imge halleri görülmektedir. Şekiller incelendiğinde Şekil 3.14.b.'de düşük aralık değerinde çıkarılan imgenin arka fonu koyu gri renk değerine sahipken, Şekil 3.14.c.'de arka fon daha açık gri tonuna sahip ve 3.14.d.'de arka fon renginin orijinal imgeye çok yakın beyaz renkte olduğu görülmektedir. Her üç imgede de mesaj bilgisinin anlaşılır olduğu görülmektedir. Geliştirilen yöntem ile bu şekilde imge içerisine mesaj gizlenebilir. Şekil 3.14.c.'de Şekil 3.14.b.'ye göre daha açık olmasının nedeni geri çıkarma işleminde gizlenen 0 ile 3 aralığındaki değere sahip bilginin geri elde edilirken orta değer seçilerek 2 bilgisi ile geri çıkartılmasıdır. Burada 0 bilgisi gizlendi ise 2 olarak geri elde edildiğinden RGB renk tonlarına göre 0'a yakın renkler koyu olduğundan koyu bir renge sahip olan veri daha açık renkte geri elde edilmiştir. Eğer 3 bilgisi gizlendi ise ve 2 olarak geri çıkarılır ve açık renk

koyu renge doğru yaklaşmaktadır. Eğer geri elde edilecek değer, orta değere yerine 0 olarak seçilirse daha koyu tonlarda bir imge elde edilir. Veya 3 olarak seçilirse daha açık tonlarda imge elde edilir. Şekil 3.14.'de imgelerin tümü incelendiğinde orijinal imge beyaz tonlara sahip olduğu için yüksek aralık seçiminde Şekil 3.14.d.'de görüldüğü gibi orijinal imgeye en yakın imge geri elde edilmektedir.



Şekil 3.14. EM-4 yöntemiyle mesaj gizleme a) EM-4 yöntemiyle gizlenen mesaj içeren orijinal imge b) düşük aralık seçilerek geri çıkarılan imge c) orta aralık seçilerek geri çıkarılan imge d) yüksek aralık seçilerek geri çıkarılan imge

Görsel olarak Şekil 3.14.a.'da verilen orijinal imgeye en yakın imge yüksek aralık ile çıkarılan Şekil 3.14.d.'de verilen imgedir. Bu bilgiyi doğrulayan sayısal veriler Tablo 3.2.'de verilmiştir. Tablo 3.2.'de Şekil 3.14.a.'daki imgenin farklı aralık değerleri ile çıkarılmış hallerinin PSNR ve MSE değerleri verilmiştir. Buna göre yüksek aralık değerindeki PSNR değeri en yüksek değere sahipken, MSE değeri ise en düşük değere sahiptir. Bu tablodan da anlaşılacağı üzere yüksek aralıkla geri çıkarılan imge orijinal imgeye en yakın değere sahiptir. Orjinal imge açık tonlar içerdiği için yüksek aralık değerli çıkarılan imge en yakın benzer imgeyi elde etmiştir.

Tablo 3.2. EM-4 yöntemiyle Şekil 3.14.a.'daki gizlenen imgenin farklı aralık değerleri ile çıkarılmış hallerinin PSNR ve MSE değerleri

PSNR	MSE	Aralık
14,55	2.276	Düşük
22,88	334	Orta
24,49	230	Yüksek

Şekil 3.15.'de siyah arka fonunda beyaz renk yazı fontu mesaj içeren imgenin gizlenmeden önceki ve farklı aralık değerleri seçilerek geri elde edilen halleri verilmiştir. Şekil 3.15.a.'da orijinal gizlenen imge, Şekil 3.15.b.'de düşük aralık seçilerek, Şekil 3.15.c.'de orta aralık seçilerek ve Şekil 3.15.d.'de yüksek aralık seçilerek çıkarılan imge halleri görülmektedir. Şekiller incelendiğinde, Şekil 3.15.b.'de düşük aralık değerinde çıkarılan imgen orijinal imgeye en yakın imge olup arka fonu koyu siyah renk değerine sahipken, Şekil 3.15.c.'de arka fon daha açık siyah tonuna sahip ve 3.15.d.'de arka fon daha da açık siyah siyah renkte olduğu görülmektedir. Her üç imgede de mesaj bilgisinin anlaşılır olduğu görülmektedir.

Şekil 3.15. EM-4 yöntemiyle mesaj gizleme a) EM-4 yöntemiyle gizlenen mesaj içeren orijinal imge b) düşük aralık seçilerek geri çıkarılan imge c) orta aralık seçilerek geri çıkarılan imge d) yüksek aralık seçilerek geri çıkarılan imge

Şekil 3.14 örneğinde orijinal imge açık renk tonlarına sahip olduğu için yüksek aralık değeri ile çıkarılan imge orijinal imgeye en yakınken, Şekil 3.15 örneğinde orijinal imge koyu renk tonlarına sahip olduğu için düşük aralık değeri ile çıkarılan imge orijinal imgeye en yakındır.

Tablo 3.3. EM-4 yöntemiyle Şekil 3.15.a.'daki gizlenen imgenin farklı aralık değerleri ile çıkarılmış hallerinin PSNR ve MSE değerleri

PSNR	MSE	Aralık
22,66	351	Düşük
18,23	977	Orta
14,79	2.157	Yüksek

Görsel olarak Şekil 3.15.a.'da verilen orijinal imgeye en yakın imge düşük aralık ile çıkarılan Şekil 3.15.b.'de verilen imgedir. Bu bilgiyi doğrulayan sayısal veriler Tablo 3.3.'de verilmiştir. Tablo 3.3.'de Şekil 3.15.a.'daki imgenin farklı aralık değerleri ile çıkarılmış hallerinin PSNR ve MSE değerleri verilmiştir. Buna göre düşük aralık değerindeki PSNR değeri en yüksek değere sahipken MSE değeri ise en düşük değere sahiptir. Bu tablodan da anlaşılacağı üzere düşük aralıkla geri çıkarılan imge orijinal imgeye en yakın değere sahiptir. Orjinal imge koyu tonlar içerdiği için düşük aralık değerli çıkarılan imge en yakın benzer imgeyi elde etmiştir.

3.2.2. 3-Bitlik yaklaşık eşleştirme alanı tabanlı tersinir veri gizleme yöntemi (EM-5)

4-bitlik yaklaşık eşletirme tabanlı yönteminden türetilen 3-bitlik yöntemde, 3-bitlik eşleştirme alanları kullanılmıştır. Gizli mesaj 3-bitlik parçalara bölünerek kodlanır. 3-bit ile onlu sistemde 8 farklı sayı oluşturulabilir. 3-bitlik bilgi, 2-bit ile temsil edilerek kodlanır. Buna göre 0 (000_2) - 1 (001_2) arası değerler 0 (00_2) ile; 2 (010_2)-3 (011_2) arası değerler 1 (01_2) ile; 4 (100_2) - 5 (101_2) arası değerler 2 (10_2) ile; ve 6 (110_2) – 7 (111_2) arası değerler ise 3 (11_2) ile kodlanır. Gizli veri ortalama değerler ile temsil edildiği için geri elde edilen imgenin görüntü kalitesi bozulmaya uğramaktadır. Ancak alıcı taraf gizli bilgiyi görüntü kalitesi düşük olsa bile algılayabilmektedir.

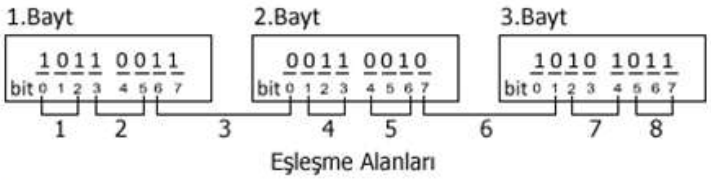
EM-5 yöntemini matematiksel olarak ifade edecek olursak; bir imgenin 2 bayt boyutunda n adet bloğa bölündüğünü ve her bloğun SB_i, i=1…n, 2 adet LSB biti bulunmaktadır. Bloklardaki bu 2-bitlik veri işaretleme amacıyla kullanılarak her bloğa 3-bitlik mesaj parçası, Mi gizlenebilir. Buna göre bir imgeye n*3 bit uzunluğunda mesaj gizlenebilir. Örneğin 600KB bir imgede 300.000 blok oluşur. Toplamda 300.000*3=900.000 bit veri gizlenebilir. 900.000/8 = 112.500 bayt/1024 = 109,9KB veri gizlenebilir.

Eşitlik 3.10.'da sırlı blokta(SB) iki bitlik LSB biti ile 3 bitlik bilginin nasıl temsil edildiği verilmiştir. Gizlenecek 3 bitlik mesaj parçası hangi eşleştirme alanı aralığına giriyorsa ona göre **SB**'deki 2-bitlik veri Eşitlik 3.10.'da gösterildiği gibi değiştirilir.

$$SB = \begin{cases} LSB(C_i) = 0, LSB(C_{i+1}) = 0 \;\; EA_1 = \{ M_i \mid 0 \leq M_i \leq 1, \;\; M_i \in Z\} \\ LSB(C_i) = 0, LSB(C_{i+1}) = 1 \;\; EA_2 = \{ M_i \mid 2 \leq M_i \leq 3, \;\; M_i \in Z\} \\ LSB(C_i) = 1, LSB(C_{i+1}) = 0 \;\; EA_3 = \{ M_i \mid 4 \leq M_i \leq 5, \;\; M_i \in Z\} \\ LSB(C_i) = 1, LSB(C_{i+1}) = 1 \;\; EA_4 = \{ M_i \mid 6 \leq M_i \leq 7, \;\; M_i \in Z\} \end{cases} \quad (3.10)$$

Eşitlik-3.10.'da **SB** sırlı bloğu, $\mathbf{C_i}$ örtü imgenin **i.** baytını, **LSB ($\mathbf{C_i}$) i.**baytın en önemsiz bitini, **EA** eşleştirme alanlarını, $\mathbf{M_i}$ gizlenecek 3-bitlik mesajın onluk karşılığını göstermektedir. Eşitlik-3.10'a göre EA_1={0,1}, EA_2={2,3}, EA_3={4,5} ve EA_4={6,7} değerlerini alır.

EM-5'nin değişim oranı $Є = \frac{1}{3} = 0{,}33$, veri gizleme kapasitesi $\alpha = \frac{3}{16} = 0{,}1875$ bulunmuştur. 16KB boyuta sahip bir imgeye 3KB veri gizleyebilir. α değeri deneysel sonuçlarda kullanılan 150 imgenin EM-5 ile veri gizleme kapasitelerinin ortalamaları baz alınarak verilmiştir.



Şekil 3.16. EM-5 yöntemi ile veri gizleme

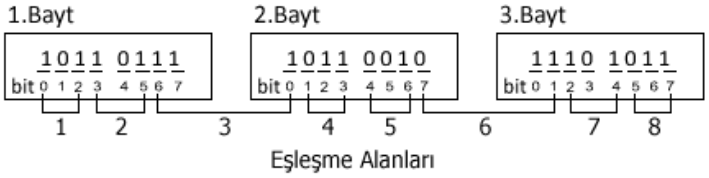
Şekil 3.16.'de EM-5 yöntemi ile 3-baytlık veri gizleme işleminde eşleştirme alanları gösterilmiştir. Buna göre 8-bit 3'e tam bölünemediği için 3.eşleşme alanı 2.baytın en anlamlı biti olan ilk bitini de kapsamaktadır. Bu durumda 3.eşleşme alanının içerdiği 110_2 bilgisi EM-5 yöntemi ile kodlanırken yaklaşık değer ile temsil edildiği için geri çıkarma aşamasında 3.baytın en anlamlı biti kaybolabilir. Bu durum geri elde edilen mesajın kalitesini oldukça bozmaktadır.

EM-5 yöntemiyle veri çıkarma işleminde aşağıdaki adımlar uygulanmaktadır.

1. Sırlı imge 2 baytlık bloklara ayrılır.
2. Sıradaki bloğun son bitleri okunarak onluk sayıya çevrilir.
3. Onluk sayı 0 ise gizli mesaj parçası 001_2 elde edilir.
4. Onluk sayı 1 ise gizli mesaj parçası 011_2 elde edilir.
5. Onluk sayı 2 ise gizli mesaj parçası 101_2 elde edilir.
6. Onluk sayı 3 ise gizli mesaj parçası 111_2 elde edilir.
7. Elde edilen mesaj parçaları birleştirilerek gizli mesaj elde edilir.

Sırlı imgenin ilk 20 biti gizli mesaj bilgisinin boyutunu göstermektedir. Gizli mesaj bilgisinin boyutu okunduktan sonra geri çıkarma işleminde de gizli mesajın boyutu kadar sırlı blok okunur. EM-5 yöntemi veriyi gizlerken yaklaşık eşleşme yöntemi ile gizlemektedir. Veri çıkarma işleminde bloktaki 2 baytlık bilginin son bitlerine gizlenen bilgiye göre gizli mesaj parçası yaklaşık olarak elde edilir.

Şekil 3.17.'de ki 3-baytlık mesaj bilgisini geri elde ederek Şekil 3.16.'daki problemi örneklendirirsek; 1.eşleşme alanı içerdiği 101_2 bilgisi EA_3 eşleştirme alanı aralığına girdiği için geri çıkarıldığında 101_2 bilgisi elde edilir. 2.eşleşme alanı içerdiği 100_2 bilgisi EA_3 eşleştirme alanı aralığına girdiği için geri çıkarıldığında 101_2 bilgisi elde edilir. 3.eşleşme alanı içerdiği 110_2 bilgisi EA_4 eşleştirme alanı aralığına girdiği için geri çıkarıldığında 111_2 bilgisi elde edilir. Burada görüldüğü gibi 3.baytın en anlamlı biti olan ilk biti 0 iken 1 olarak değişmiştir. Bu durumda 3.baytın sayısal değeri 128 artmıştır. Bu durumda gizlenen mesaj eğer bir imge ise kalitesini oldukça düşürecektir.



Şekil 3.17. EM-5 yöntemi ile veriyi geri elde etme

Şekil 3.17.'de Şekil 3.16.'da verilen EM-5 yöntemiyle gizlenmiş 3-baytlık bilginin EM-5 yöntemi ile geri elde edilmesi sonucunda oluşan 3-baytlık veri verilmiştir. EM-5 yöntemi ile veri gizlemesi yapılmadan önce 3-bayt sırasıyla 179, 50, 171 onluk değerlere sahipken geri elde işlemi sonucunda elde edilen 3-bayt veri sırasıyla 183,178, 235 onluk değerlere sahiptir. Özellikle 2.bayttaki veri oldukça değişmiştir.

Bu durumun gizlenen imge üzerindeki yansıması Şekil 3.18.'de verilmiştir.

a) b)

Şekil 3.18. EM-5 yöntemiyle veri gizleme a) EM-5 yöntemiyle gizlenen imge b) EM-5 yöntemiyle geri çıkarılan

Şekil 3.18.'de verilen imgelere bakıldığında EM-5 yöntemiyle gizlenmiş kedi imgesinin geri çıkarılmış halinden bozulma görülmektedir. Sonuç olarak karşı tarafa iletilen imge anlaşılsa da imge kalitesi oldukça bozulmuştur.

3.2.3. 2-Bitlik yaklaşık eşleştirme alanı tabanlı tersinir veri gizleme yöntemi (EM-6)

2-bitlik yaklaşık eşletirme alanlı tabanlı veri gizleme yöntemi 4-bitlik yaklaşık eşleştirme alanı tabanlı veri gizleme yönteminden türetilmiştir. Gizli mesaj 2-bitlik parçalara bölünerek kodlanır. 2-bit ile onlu sistemde 4 farklı sayı kodlanabilir. 2-bitlik veri 1-bit ile temsil edilerek kodlanır. Buna göre 0 (00_2) - 1 (01_2) arası değerler 0 (0_2) ile; 2 (10_2)-3 (11_2) arası değerler 1 (1_2) ile; kodlanır. Gizli bilgideki

2-bitlik değerler 1-bit ile kodlandığı için 4 farklı durum 2 duruma düşmektedir. Buda alıcı tarafda elde edilen görüntü kalitesini olumsuz yönde etkilemektedir. Fakat alıcı tarafta elde edilen mesaj anlaşılmayacak kadar bozulmamaktadır.

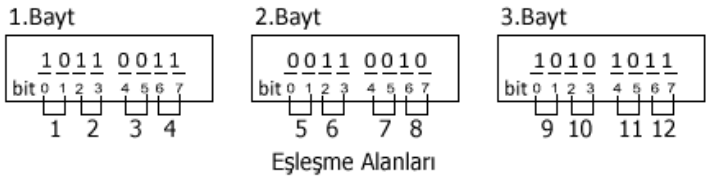
EM-6 yöntemini matematiksel olarak ifade edecek olursak; bir imge 1-bayt boyutunda **n** adet bloğa bölünür ve her bloğun **SB_i, i=1…n,** 1 adet LSB biti bulunmaktadır. Bloklardaki bu 1-bitlik veri işaretleme amacıyla kullanılarak her bloğa 2-bitlik mesaj parçası **(M_i)** gizlenebilir. Buna göre bir imgeye **n*2** bit uzunluğunda mesaj gizlenebilir. Örneğin; 500x400 boyutlarında renkli bir imgede 500*400*3=600.000 blok oluşur. Toplamda 600.000*2=1.200.000 bit veri gizlenebilir. 1.200.000/8 = 150.000 bayt/1024 = 146,5KB veri gizlenebilir. Böylece EM-6 yöntemiyle, EM-4 yöntemiyle aynı oranda veri gizlemektir. Bunun nedeni EM-4 yöntemi 2-bitlik alana 4-bit mesaj parçası gizlerken EM-6 yöntemi 1-bitlik alana 2-bit mesaj gizlediği için aynı oranda mesaj gizlemektedirler.

Eşitlik 3.12.'de sırlı blokta (SB) 1-bitlik LSB biti ile 2-bit bilginin nasıl temsil edildiği verilmiştir. Gizlenecek 2-bitlik mesaj parçası hangi eşleştirme alanı aralığına giriyorsa ona göre **SB**'deki 2-bitlik veri Eşitlik 3.12.'da gösterildiği gibi değiştirilir.

$$\text{SB}=\begin{cases} LSB(C_i)=0,\ \ EA_1=\{\,M_i \mid 0\le M_i\le 1,\ M_i\in Z\} \\ LSB(C_i)=1,\ \ EA_2=\{\,M_i \mid 2\le M_i\le 3,\ M_i\in Z\} \end{cases} \tag{3.12}$$

Eşitlik-3.12.'de **SB** sırlı bloğu, $\mathbf{C_i}$ örtü imgenin **i.** baytını, **LSB ($\mathbf{C_i}$) i.**baytın en önemsiz bitini, **EA** eşleştirme alanlarını, M_i gizlenecek 2-bitlik mesaj bitinin onluk karşılığını göstermektedir. Eşitlik-3.12.'e göre EA_1={0,1}, EA_2={2,3} değerlerini alır.

EM-6'nın değişim oranı $Є = \frac{1}{4}$ = 0,25 veri gizleme kapasitesi $\alpha=\frac{1}{4}$=0,25 bulunmuştur. Örneğin, 60KB boyuta sahip bir imgeye 15KB veri gizleyebilir. Є ve α değerleri deneysel sonuçlarda kullanılan 150 imgenin EM-6 ile veri gizleme kapasitelerinin ortalamaları baz alınarak verilmiştir.



Şekil 3.19. EM-6 yöntemi ile veri gizleme

EM-6 yönteminde 2-bitlik eşleştirme alanları kullanıldığından dolayı Şekil 3.19.'da görüldüğü gibi 3-baytlık bilgide 12 adet eşleşme alanı oluşmaktadır. 2-bitlik yöntemde veri gizleme işleminde gizlenen baytın MSB bitleri 2 eşleşme alanına bölündüğü için geri dönüşüm işleminde önemli veri kayıpları oluşabilmektedir. Gizli bilgideki 2-bitlik değerler 1-bit ile kodlandığı için 4 farklı durum 2 duruma düşmektedir. MSB bitlerindeki kayıplar büyük değişimlere neden olacağı için geri elde edilen mesajın kalitesini oldukça bozmaktadır.

EM-6 yönteminin veri gizleme algoritması EM-4 yöntemiyle aynı şekildedir. EM-4 yönteminde 4-bitlik eşleştirme alanı kullanılırken EM-6 yönteminde 2-bitlik eşleştirme alanı kullanılmıştır. EM-6 yöntemiyle veri çıkarma işleminde aşağıdaki adımlar uygulanmaktadır.

1. Sırlı imge 1 baytlık bloklara ayrılır.
2. Sıradaki bloğun LSB biti 0 ise blokta gizli mesaj parçası 01_2 olarak elde edilir.
3. Sıradaki bloğun LSB biti 1 ise blokta gizli mesaj parçası 11_2 olarak elde edilir.
4. Elde edilen mesaj parçaları birleştirilerek gizli mesaj elde edilir.

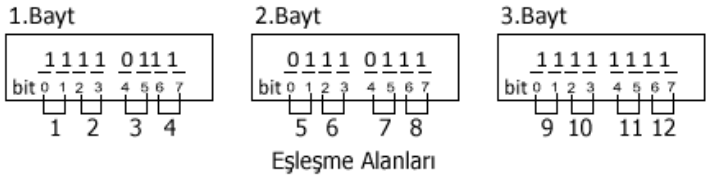
Sırlı imgenin ilk 20 biti gizli mesaj bilgisinin boyutunu göstermektedir. Gizli mesaj bilgisinin boyutu okunduktan sonra geri çıkarma işleminde de gizli mesajın boyutu kadar sırlı blok okunur. EM-6 yöntemi veriyi gizlerken yaklaşık eşleşme yöntemi ile gizlemektedir. Veri çıkarma işleminde bloktaki 1 baytlık bilginin son bitine gizlenen bilgiye göre gizli mesaj parçası yaklaşık olarak elde edilir. Geri elde edilen 2-bitlik mesaj parçası eşleştirme alanındaki değerlein orta noktası baz alınarak

seçilmiştir. Örneğin EA_1={0,1} değerlerine sahiptir ve orta değer olmadığı için geri elde edilen veri olarak 1 seçilmiştir.

EM-6 yöntemiyle gizlenen Şekil 3.18.'deki 3-baytlık mesaj verisini geri elde edersek; 1.eşleşme alanı içerdiği 10_2 verisi, EA_2 eşleştirme alanı aralığına girdiği için geri çıkarıldığında 11_2 verisi elde edilir. 1.eşleşme alanı 1.baytın en önemli iki biti olduğu için buradaki bir değişim büyük kayıplara yol açabilir. Örneğimizde gerçekte gizlenen 10_2 verisi yerine 11_2 verisi geri elde edilmiştir. Bu durumda 1.baytın sayısal değeri 128 artmıştır. Bu durumda geri elde edilen gizli imgenin kalitesini oldukça düşürecektir. 2.eşleşme alanı içerdiği 11_2 verisi EA_2 eşleştirme alanı aralığına girdiği için geri çıkarıldığında 11_2 verisi elde edilir. 3.eşleşme alanı içerdiği 00_2 bilgisi EA_1 eşleştirme alanı aralığına girdiği için geri çıkarıldığında 01_2 bilgisi elde edilir. Bu şekilde tüm veri gizlenir.

Şekil 3.20.'de Şekil 3.19.'da verilen EM-6 yöntemiyle gizlenmiş 3-baytlık verinin EM-6 yöntemi ile geri elde edilmesi sonucunda oluşan 3-baytlık veri verilmiştir. EM-6 yöntemi ile veri gizlemesi yapılmadan önce 3-bayt sırasıyla 179, 50, 171 onluk değerlere sahipken geri elde işlemi sonucunda elde edilen 3-bayt veri sırasıyla 247,119, 255 onluk değerlere sahiptir. Özellikle 3.bayttaki veri oldukça değişmiştir.

Bu durumun gizlenen imge üzerindeki yansıması Şekil 3.20.'de verilmiştir.



Şekil 3.20. EM-6 yöntemi ile veriyi geri elde etme

a) b)

Şekil 3.21. EM-6 yöntemiyle veri gizleme a) EM-6 yöntemiyle gizlenen imge b) EM-6 yöntemiyle geri çıkarılan imge.

Şekil 3.21.'de verilen imgelere bakıldığında EM-6 yöntemiyle gizlenmiş kedi imgesinin geri çıkarılan verilerinde bozulma görülmektedir. Sonuç olarak karşı tarafa iletilen imge anlaşılsa da imge kalitesi EM-5 yöntemine göre daha çok bozulmuştur.

Bu bölümle tez çalışmasında geliştirilen 6 sırtörtme yönteminin çalışma yapısı algoritmaları ve akış şemaları detaylarıyla verilmiştir. Her yöntemin birbirine göre üstünlük ve zayıflıkları incelenerek kıyaslamalar yapılmıştır. Bir sonraki bölümde tez çalışmasında geliştirilen sırörtme yöntemlerinin deneysel sonuçları sayılarla ve grafiklerle verilecektir. Ayrıca literatüdeki eşdeğer çalışmalar ile de başarım performansı kıyaslaması yapılacaktır.

BÖLÜM 4. DENEYSEL SONUÇLAR VE LİTERATÜRLE KARŞILAŞTIRMA

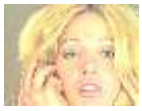
Tez çalışmasında yeni bir veri gizleme yöntemi ve bu yöntemin türevleri geliştirilmiştir. İyi bir veri gizleme yöntemi, ilk olarak veri gizleme işlemi sonucunda örtü imgede gözle görülebilecek bozulmalara sebebiyet vermemesi gerekmektedir. Ayrıca belirli imge ölçüt analizlerinden iyi sonuçlar üretmesi gerekmektedir. Bununla birlikte veri gizleme işleminin sıraçma yöntemlerince de algılanamaması yöntemin güvenliği için önemlidir. Geliştirilen yöntemlerden ilk bulunan üç yöntem EM-1, EM-2 ve EM-3 yöntemleri gizli bilgiyi geri çıkarma aşamasında orjinal dosyaya ihtiyaç duymaktadır. İlk üç yöntem baz alınarak geliştirilen EM-4, EM-5 ve EM-6 yöntemleri ise veri çıkarma aşamasında orjinal dosyaya ihtiyaç duymamaktadır. Karşılaştırmalarda kullanılan literatürdeki çalışmalarda geri çıkarma işlemi sırasında orjinal dosyaya ihtiyaç duymamaktadır. Aynı şartlar altında kıyaslama yapabilmek için ilk üç yöntem test sonuçlarında yer almamaktadır. EM-4, EM-5, EM-6 yöntemlerinin başarım performanslarını ölçmek için parametrik analiz yöntemi ile literatürde sıklıkla kullanılan ortalama karesel hata MSE (Mean Square Error) ve tepe sinyal gürültü oranı PSNR (Peak Signal to Noise Ratio) imge ölçütleri kullanılmıştır. Geliştirilen yöntemlerin sıraçma algoritmalarına karşı dayanıklılığını ölçebilmek için ki-kare, komşuluk histogramı (neighbourhood histogram) ve piksel farkı histogramı (pixel difference histogram) yöntemleri kullanılmış ve sonuçları verilmiştir. Ayrıca geliştirilen yöntemlerin görsel analizlere karşı dayanıklılığını ölçebilmek için piksel atağı ve LSB saldırıları kullanılmıştır. Yöntemlerin başarımı literatürdeki eşdeğer çalışmalar ile de kıyaslanmıştır.

4.1. Kullanılan Test Seti

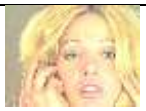
Geliştirilen sırörtme yöntemlerinin başarımını farklı parametrelere göre test etmek için Tablo 4.1.'de verilen literatürde sıklıkla kullanılan örtü imgeler kullanışmıştır.

Ayrıca başarım sonuçlarını kesin olarak ortaya koyabilmek için 150 adet 500x400 boyutlarında renkli imgeler kullanılmıştır. İmgeler Internet üzerinden indirilmiş uçak, hayvan, çiçek ve balık görselleri içermektedir. Kullanılan imgelere ve geliştirilen yöntemlerin Matlab® kodlarına Internet üzerinden erişilebilir [63].

Tablo 4.1. Tez çalışmasında kullanılan farklı parametrelere göre örtü imgeler.

Tür	İmge Boyut	İçerik	İmge
JPEG	640x480	Karmaşık Renkli	
JPEG	400x300	Karmaşık Renkli	
JPEG	300x250	Karmaşık Renkli	
JPEG	640x480	Düz Renkli	
JPEG	400x300	Düz Renkli	
JPEG	300x250	Düz Renkli	
BMP	640x480	Karmaşık Renkli	
BMP	400x300	Karmaşık Renkli	
BMP	300x250	Karmaşık Renkli	
BMP	640x480	Düz Renkli	

Tablo 4.1. (Devam)

Tür	İmge Boyut	İçerik	İmge
BMP	400x300	Düz Renkli	
BMP	300x250	Düz Renkli	

4.2. Görsel Test

Geliştirilen sırörtme yöntemlerinin gizleme işlemi sonrasında örtü imgede gözle görülebilecek bozulmalara sebebiyet vermemesi gerekmektedir. Gizlenen verilerin boyutlarında Eşitlik-4.1.'de verilen piksel başına bit BPP (Bit Per Pixel) ölçü birimi kullanılmıştır.

$$BPP = \frac{Gizlenen\ bit\ sayısı}{Toplam\ piksel\ sayısı} \tag{4.1}$$

Şekil 4.1.'de test için kullanılan 150 imgeden örnek bir imgeye önerilen gizleme yöntemleri ile 1,5 bpp oranında veri gizlenmiş sırlı imgeler verilmiştir. İmgelerde gözle algılanabilecek bozulmalar mevcut değildir.

a) b)

Şekil 4.1. Test setinden örnek bir imgenin önerilen yöntemlerde 1,5 bpp (112KB) veri gizlenmiş sırlı görünümleri

c) d)

Şekil 4.1. (Devam)

4.3. İmge ölçütleri

Literatürdeki çalışmaların bir çoğunda sıklıkla kullanılan örtü imgelerin veri gizleme işleminden sonra bozulmalarını tespit edebilmek için örtü imgeyi orijinal imge ile kıyaslama yapan ölçütler mevcuttur. Örtü imgelerdeki bozulmaları ölçebilmek için literatürde sıklıkla kullanılan Eşitlik-4.2.'de verilen ortalama karesel hata (MSE) ve Eşitlik-4.3.'de verilen tepe sinyal gürültü oranı (PSNR) kalite ölçütleri kullanılmıştır. Ayrıca histogram tabanlı algısal kalite ölçütü (HPQA) [64], renkli imge kalite ölçütü (CQM) [65], yapısal benzerlik kalite ölçütü (SSIM) [66] ve evrensel kalite indeksi (UQI) [67] kalite ölçütüde kullanılmıştır.

Eşitlik-4.2.'de m ve n imgenin satır ve sutun bilgilerini; O orjinal örtü imgeyi; S ise sırlı imgeyi temsil etmektedir. MSE değeri hesaplandıktan sonra bir sonraki adım PSNR hesaplamasıdır. PSNR örtü imge ile sırlı imge arasındaki benzerlik oranını ölçen bir imge ölçütüdür. Eşitlik-4.3.'de MAX bir pikselin alabileceği maksimum değer ve genellikle 255'dir.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[O(i,j) - S(i,j)]^2 \tag{4.2}$$

$$PSNR = 10log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{4.3}$$

4.4. Önerilen Yöntemlerin Parametrik Başarım Analizi

Tez çalışmasında geliştirlen EM-4, EM-5 ve EM-6 yöntemleri ile literatürdeki çalışmaların başarımlarını kıyaslayabilmek için parametrik analiz yapılmıştır. Parametrik analizde 640x480, 400x300 ve 300x250 boyutlarında Tablo 4.1.'de özellikleri verilen literatürde sıklıkla kullanılan 6 JPEG imge ile 6 BMP imge kullanılmıştır. JPEG ve BMP imgelerden ilk üçü karmaşık renkler içerirken son üçü ise düz renkler içermektedir. Parametrik analizde örtü imge boyutu, imge kayıt türü, imge içeriği ve gizlenen imgenin boyutu olmak üzere 4 farklı paremetre çapraz eşleştirilerek kullanılmıştır. Toplam 12 imgeye 0,5bpp ve 1bpp olmak üzere iki farklı oranda imge verisi EM-4, EM-5, EM-6 ve literatürdeki iki yöntem Mielkainen ve Chan olmak üzere 5 farklı yöntemle gizlenerek 120 sırlı imge elde edilmiştir. 120 sırlı imgenin PSNR değerleri hesaplanarak Tablo 4.2.'de verilmiştir. Tabloda verilen sonuçlar PSNR değerlerine göre en başarılı yöntemden en az başarılı yönteme göre sıralanmıştır. Koyu olarak işaretlenen EM-4 yöntemi kıyaslamaya katılan yöntemlerden en başarılı yöntem olduğu tespit edilmiştir. Tez çalışmasında geliştirilen üç yöntem de literatürdeki yöntemlerden daha başarılı sonuçlar elde etmiştir.

640x480 JPEG düz renkler içeren imgede 0,5bpp gizleme oranında EM-4 yöntemi ile 66,2db PSNR değeri elde edilirken EM-6 66,2db, EM-5 65,0db, Chan 64,8db ve Mielkainen yöntemi ise 64,4db PSNR değeri elde etmiştir. Buna göre EM-4 yönteminin aynı oranda veriyi gizlerken örtü imge de en az değişiklik yaptığı görülmektedir. EM-6 yöntemi de EM-4 yöntemi ile aynı gizleme oranlarını kullanmasından dolayı EM-4 yöntemine en yakın PSNR değerlerini vermesine rağmen geri elde edilen imgenin kalitesini oldukça bozmaktadır. Chan ile EM-4 yöntemi arasında 1,4db, Mielkainen ile EM-4 arasında 1,8db PSNR farkı bulunmaktadır. Bu fark oldukça yüksektir. EM-4 yöntemi 2-bitlik alana 4 bit veri gizlediği için iki yönteme göre örtü imgede yarı yarıya daha az değişiklik yapmaktadır.

640x480 JPEG düz rekler içeren imgede imgenin alabileceği maksimum oranda 1bpp ile gizleme yapıldığında EM-4 63,2dB, EM-6 63,2dB, EM-5 61,9dB, Chan 61,9dB ve Mielkainen 61,4dB PSNR değeri elde etmiştir. Bu durumda da en başarılı sonucu EM-4 yöntemi vermiştir. 0,5bpp gizleme oranında olduğu gibi Chan ile EM-4 yöntemi arasında 1,3dB, Mielkainen ile EM-4 arasında 1,8db yüksek oranda PSNR farkı bulunmaktadır.

İmge içeriğine göre düz ve karmaşık renkler içeren imgelerde EM-4 yöntemi ile elde edilen PSNR değerleri karşılaştırıldığında en yüksek farkın 0,2dB oranında karmaşık renkler içeren 300x250 boyutlarında imge ve 0,5bpp gizleme oranında olduğu tespit edilmiştir. Karmaşık imgede düz imgeye göre daha düşük PSNR değeri elde edilmiştir. Karmaşık imgeler renklerin çeşitliliğinden dolayı bit dizilişlerinde de çeşitlilik içermektedir. Bundan dolayı her baytın son bitleri de oldukça çeşitlilik içermektedir. Veri gizlerken gizlenecek 1-bitlik verinin LSB ile aynı olması durumunda veri değişikliği oluşmamaktadır. Karmaşık imgelerde bu durum düz imgelere göre daha çok bit değişikliği ile sonuçlanmaktadır ve imge içeriği başarımı etkilemektedir.

İmge boyutlarına göre EM-4 yönteminin performansı analiz edildiğinde, imge boyutlarının artması veya azalması elde edilen PSNR değerlerinde çok fazla farklılık oluşturmamıştır. Buradan da anlaşıldığı üzere imge boyutu başarımı etkilememektedir.

İmge türüne göre EM-4 yönteminin başarımı analiz edildiğinde, imge türünün JPEG yada BMP olması elde edilen PSNR değerlerinde neredeyse hiç farklılık oluşturmamıştır. Bunun nedeni, sıkıştırılmış JPEG imgelerde gizleme yapılırken MATLAB®, JPEG imgeyi BMP formatında olduğu gibi ham haline dönüştürerek açmaktadır. JPEG imge BMP seviyesine getirilerek işlendiği için BMP ile aralarında çok fazla fark oluşmamaktadır ve imge kayıt türü başarımı etkilememektedir.

Tablo 4.2. Tez çalışmasında geliştirilen ve literatürdeki yöntemlerin parametrik analizi ile elde edilmiş PSNR (db) değerleri

	İmge içeriği	**Düz**						**Karmaşık**					
	Boyut	640x480		400x300		300x250		640x480		400x300		300x250	
	Bpp	0,5	1	0,5	1	0,5	1	0,5	1	0,5	1	0,5	1
JPEG	**EM-4**	**66,2**	**63,2**	**66,2**	**63,2**	**66,3**	**63,2**	**66,2**	**63,2**	**66,2**	**63,3**	**66,1**	**63,2**
	EM-6	66,2	63,2	66,2	63,2	66,2	63,2	66,2	63,2	66,2	63,2	66,1	63,2
	EM-5	65,0	61,9	65,0	62,0	64,9	61,9	65,0	61,9	65,0	62,1	65,0	62,0
	Chan [3]	64,8	61,9	64,8	61,9	65,1	62,0	64,9	61,8	64,6	61,7	64,9	61,8
	Mielkainen [2]	64,4	61,4	64,4	61,4	64,4	61,4	64,4	61,4	64,7	61,9	64,4	61,4
BMP	**EM-4**	**66,2**	**63,1**	**66,2**	**63,2**	**66,3**	**63,2**	**66,2**	**63,2**	**66,2**	**63,3**	**66,2**	**63,1**
	EM-6	66,2	63,2	66,2	63,2	66,2	63,2	66,2	63,2	66,2	63,2	66,2	63,2
	EM-5	65,0	61,9	65,0	62,0	64,9	61,9	64,9	61,9	64,9	62,1	64,9	61,9
	Chan [3]	64,9	61,9	64,8	61,8	65,1	62,1	64,9	61,7	64,6	61,7	64,9	61,8
	Mielkainen [2]	64,4	61,4	64,4	61,4	64,4	61,4	64,4	61,4	64,7	61,8	64,4	61,4

Parametrik analizin bir diğer sonucu MSE değerleridir. MSE karesel hata değerlerinde PSNR değerlerine ters orantılı olarak yöntemin başarımı yükseldikçe MSE değeri sıfıra yaklaşır. Tablo 4.3.'de parametrik analiz sonucu elde edilen MSE değerleri verilmiştir. Buna göre EM-4 yöntemi, en düşük hata değerlerini elde ederek yine en başarılı yöntem olarak tespit edilmiştir. EM-6 yöntemi, PSNR sonuçlarında olduğu gibi EM-4 yöntemine çok yakın değerler elde etmiştir. EM-5 yöntemi de Chan ve Mielkainen yönteminden daha başarılı sonuçlar elde etmiştir.

İmge içeriğine göre düz ve karmaşık renkler içeren imgelerde EM-4 yöntemi ile elde edilen MSE değerleri karşılaştırıldığında en yüksek farkın 0,2 oranında 400x300 boyutlarında imge ve 1bpp gizleme oranında olduğu tespit edilmiştir. Karmaşık imgede düz imgeye göre daha yüksek MSE değeri elde edilmiştir ve bu değerler PSNR değerlerini doğrulamaktadır.

İmge boyutlarına göre EM-4 yönteminin başarımında, imge boyutlarının artıp veya azalması elde edilen MSE değerlerinde çok fazla farklılık oluşturmamıştır.

Tablo 4.3. Tez çalışmasında geliştirilen yöntemler ve literatürdeki parametrik analizi ile elde edilmiş MSE değerleri

	İmge içeriği	**Düz**						**Karmaşık**					
	Boyut	640x480		400x300		300x250		640x480		400x300		300x250	
	Bpp	0,5	1	0,5	1	0,5	1	0,5	1	0,5	1	0,5	1
JPEG	**EM-4**	**0,015**	**0,031**	**0,015**	**0,031**	**0,015**	**0,030**	**0,015**	**0,031**	**0,015**	**0,029**	**0,015**	**0,031**
	EM-6	0,016	0,031	0,016	0,031	0,016	0,031	0,016	0,031	0,016	0,031	0,016	0,031
	EM-5	0,021	0,042	0,021	0,041	0,021	0,042	0,021	0,042	0,021	0,040	0,020	0,041
	Chan [3]	0,021	0,044	0,022	0,044	0,020	0,042	0,021	0,044	0,024	0,046	0,022	0,045
	Mielkainen[2]	0,024	0,047	0,024	0,047	0,023	0,047	0,024	0,047	0,022	0,042	0,024	0,047
BMP	**EM-4**	**0,015**	**0,031**	**0,015**	**0,031**	**0,015**	**0,030**	**0,015**	**0,031**	**0,015**	**0,029**	**0,015**	**0,031**
	EM-6	0,016	0,031	0,016	0,031	0,016	0,031	0,016	0,031	0,016	0,031	0,015	0,031
	EM-5	0,021	0,042	0,021	0,041	0,021	0,042	0,021	0,042	0,021	0,040	0,021	0,042
	Chan [3]	0,021	0,044	0,021	0,044	0,019	0,042	0,021	0,044	0,021	0,044	0,021	0,044
	Mielkainen[2]	0,024	0,047	0,024	0,047	0,023	0,047	0,023	0,047	0,022	0,042	0,024	0,047

İmge türüne göre EM-4 yönteminin başarım analizinde, PSNR sonuçlarında olduğu gibi imge türünün JPEG ya da BMP olması elde edilen MSE değerlerinde neredeyse hiç farklılık oluşturmamıştır.

Parametrik analizde ölçülen değerlerden birisi de değişen bit sayısıdır. Aynı oranda veri gizleyerek örtü imgede en az bit değiştiren yöntem, en başarılı yöntemdir. Tablo 4.4.'de parametrik analiz sonucu elde edilen değişen bit sayıları verilmiştir. Buna göre EM-4 yöntemi en az bit değiştirdiği için yine en başarılı yöntem olarak tespit edilmiştir. EM-6 yöntemi diğer sonuçlarda olduğu gibi EM-4 yöntemine çok yakın değerler elde etmiştir. EM-5 yöntemi de Chan ve Mielkainen yönteminden daha başarılı sonuçlar elde etmiştir.

Tablo 4.4. Tez çalışmasında geliştirilen yöntemler ve literatürdeki parametrik analizi ile elde edilmiş değişen bit sayıları

	İmge içeriği	**Düz**						**Karmaşık**					
	Boyut	640x480		400x300		300x250		640x480		400x300		300x250	
	Bpp	0,5	1	0,5	1	0,5	1	0,5	1	0,5	1	0,5	1
JPEG	**EM-4**	**14257**	**28685**	**5557**	**11170**	**3428**	**6911**	**14255**	**28591**	**5605**	**10730**	**3564**	**7064**
	EM-6	14453	28907	5638	11202	3493	7008	14393	28730	5618	11156	3556	7057
	EM-5	19049	38362	7388	14870	4709	9365	19131	38512	7457	14484	4582	9282
	Chan	20210	40489	7185	14456	3845	8142	19745	41014	7456	14456	4145	9457
	Mielkainen	21722	43341	8451	16967	5271	10571	21719	43346	7927	15255	5308	10506
BMP	**EM-4**	**14270**	**28707**	**5615**	**11197**	**3404**	**6885**	**14394**	**28734**	**5618**	**10756**	**3511**	**7099**
	EM-6	14442	28885	5626	11161	3491	6999	14362	28650	5625	11186	3480	7029
	EM-5	18992	38372	7396	14832	4707	9375	19224	38498	7502	14519	4685	7099
	Chan	20197	40674	7410	14978	4745	9087	20978	41174	7698	15104	4687	9425
	Mielkainen	21732	43369	8463	16922	5265	10557	21597	43117	7946	15298	5343	10616

İmge içeriğine göre düz ve karmaşık renkler içeren imgelerde EM-4 yöntemi ile elde edilen değişen bit sayıları karşılaştırıldığında en yüksek farkın 440 değerinde, karmaşık renkler içeren 400x300 boyutlarında imge ve 1bpp gizleme oranında olduğu tespit edilmiştir. Karmaşık imgede düz imgeye göre daha fazla bit değişikliği olmuştur. Böylece MSE ve PSNR değerlerini doğrulayan sonuçlar elde edilmiştir.

İmge boyutlarına göre EM-4 yönteminin başarımı analiz edildiğinde ve imge boyutlarının azalıp ya da artması ile elde edilen değişen bit sayılarına bakıldığında; büyük imge küçük imgeye göre daha fazla veri gizleneceğinden dolayı ölçüt olarak kullanılmamıştır.

İmge türüne göre EM-4 yönteminin başarımı analiz edildiğinde ve imge türünün JPEG yada BMP olması elde edilen değişen bit sayılarına bakıldığında en yüksek farkın 143 değerinde karmaşık renkler içeren 640x480 boyutlarında imge ve 1bpp

gizleme oranında olduğu tespit edilmiştir. 921.600 bit gizlenen veri boyutuna göre bu kadar fark çok küçük bir değerdir.

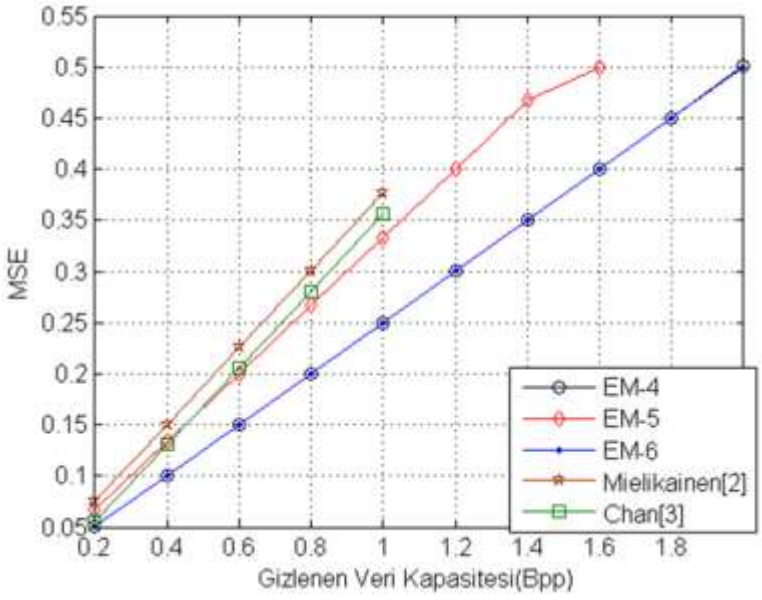
4.5. Önerilen Yöntemlerin Ortalama Başarım Analizi

Geliştirilen sırörtme yöntemlerinin veri gizleme başarımlarını ölçebilmek ve literatürdeki eşdeğer iki çalışma ile kıyaslayabilmek için test setimizdeki 150 imgeye tez çalışmasında geliştirilen yöntemlerden EM-4, EM-5 ve EM-6 ile ve literatürdeki eşdeğer iki çalışma ile 15KB (0.2bpp) ve katları olacak şekilde 10 kez maksimum 150KB (2bpp) veri gizlenmiştir. Tez çalışmasında geliştirilen EM-1, EM-2 ve EM-3 yöntemleri gizli veriyi çıkarırken orijinal dosyaya ihtiyaç duyduğu için karşılaştırmaya alınmamıştır. Literatürdeki çalışmalar ve tez çalışmasında geliştirilen EM-4, EM-5 ve EM-6 yöntemleri gizli veriyi çıkarırken orijinal dosyaya ihtiyaç duymayan yöntemlerdir. Her gizleme sonrasında oluşan sırlı imgenin MSE değerleri hesaplanmıştır. Her yöntem için hesaplanan MSE değerlerinin ortalamasına ait grafik Şekil 4.2.'de verilmiştir.

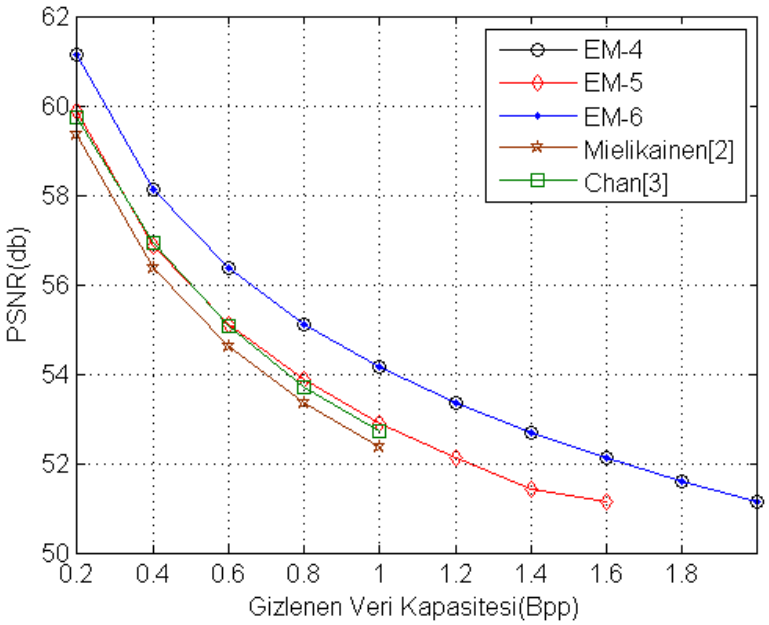
150 imgeye 0 ile 2 bpp arasında önerilen yöntemler ve literatürdeki yöntemlerle veriler gizlenmiştir. Her gizleme yönteminde imgelerdeki bozulmalar Eşitlik-4.2, Eşitlik-4.3.'de verilen MSE ve PSNR ölçüm parametreleri ile ölçülmüştür. Literatürdeki Mielkainen ve Chan yöntemi dosyaya en fazla dosya boyutunun $\frac{1}{8}$'i kadar oranda veri gizleyebildiği için maksimum 75KB (1bpp) veri gizleyebilmektedir. Ancak, EM-4, EM-5 ve EM-6 yöntemleri literatürdeki yöntemlere göre daha fazla veri gizleyebilmektedir. Şekil 4.2.'deki grafiğe göre EM-4 yönteminin 5 yöntemden en başarılı sonucu veren yöntem olduğu görülmektedir. Bu da örtü imgede en az değişiklik yaptığını doğrulamaktadır. EM-4 yöntemi, 2-bitlik değişimle 4-bitlik veri gizlediği için Şekil 4.2.'yi doğrulayan şekilde en az bit değiştirmiştir. EM-6 yöntemi 1 bitlik değişimle 2-bit veri gizlediği için EM-4 yöntemiyle neredeyse aynı sonuçlarla veri gizlemiş ve grafikde sonuçlar üst üste çakışmıştır. EM-5 yöntemi 2-bitlik alana 3-bit veri gizlediği için üçüncü sırada en az değişiklikle veri gizlemiştir. Chan yöntemi EM-5 yöntemine en yakın sonuçları veren yöntemdir. Şekil-4.2.'e göre başarı sıralaması EM-4, EM-6, EM-5, Chan,

Mielikainen şeklinde olmuştur. 3 önerilen yöntemden hepsi literatürdeki iki çalışmadan daha az değişiklikle veri gizlemeyi başarmıştır.

Veri gizlenmiş sırlı 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 sırörtme yönteminin örtü imgede daha az değişiklik yaptığını göstermektedir. Şekil 4.3.'de 150 sırlı imgenin PSNR ölçüm değerlerinin ortalamalarını gösteren grafik verilmiştir. Grafiğe göre PSNR değeri en yüksek çıkan EM-4, 5 yöntemden en başarılı sonucu veren yöntem olduğu görülmektedir. Bu da örtü imgede en az değişiklik yaptığını doğrulamaktadır. EM-4 yönteminden sonra en yüksek PSNR değerine sahip diğer yöntem EM-6 yöntemidir. Şekil 4.3.'e göre PSNR değerleri yüksekten düşüğe göre EM-4, EM-6, EM-5, Chan, Mielikainen, şeklinde sıralanmaktadır. Bu sonuca göre önerilen üç yöntemden tümü literatürdeki iki çalışmadan daha az değişiklikle veri gizlemeyi başarmıştır. Şekil 4.3.'e bakıldığında PSNR değerlerine görede örtü imgesinde en az değişiklik yapan yöntemlerin sıralaması Şekil 4.2 ile aynıdır.

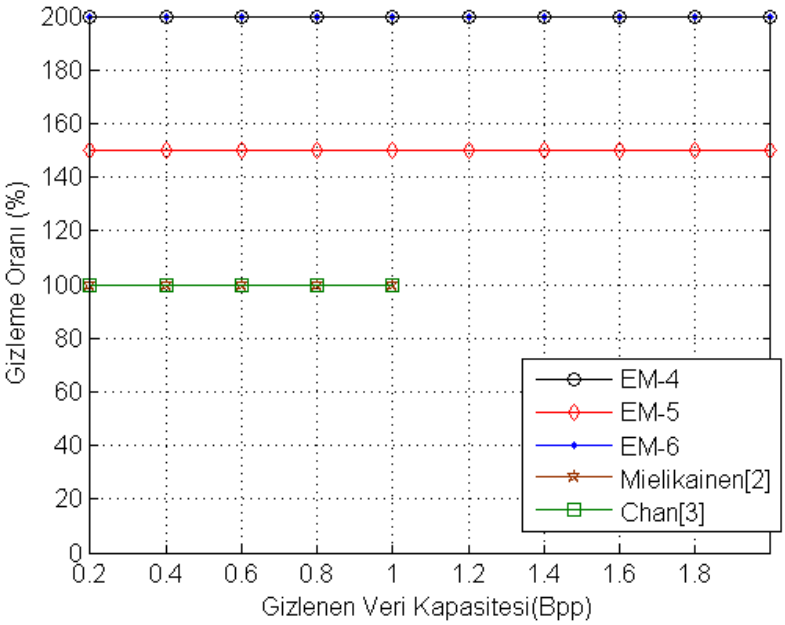


Şekil 4.2. 150 imgeye önerilen yöntemler ve literatür çalışmalarıyla farklı oranlarda veri gizlemeleriyle oluşan sırlı imgeler için MSE değerlerinin ortalamalarının karşılaştırılması



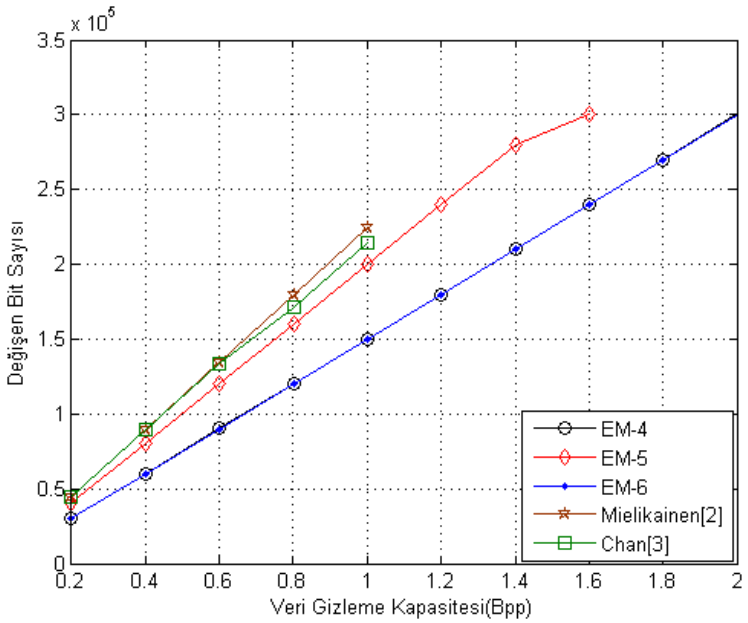
Şekil 4.3. 150 imgeye önerilen yöntemler ve literatür çalışmalarıyla farklı oranlarda veri gizlemeleriyle oluşan sırlı imgeler için PSNR değerlerinin ortalamalarının karşılaştırılması

Geliştirilen sırörtme yöntemlerinin veri kapasitelerini ölçebilmek ve literatürdeki eşdeğer iki çalışma ile kıyaslayabilmek için test setimizdeki 150 imgeye tez çalışmasında geliştirilen yöntemlerden EM-4, EM-5 ve EM-6 ile ve literatürdeki eşdeğer iki çalışma ile 15KB (0.2bpp) ve katları olacak şekilde 10 kez maksimum 150KB (2bpp) veri gizlenmiştir. Her gizleme sonrasında oluşan sırlı imgenin veri kapasiteleri Eşitlik 3.7.'e göre hesaplanmıştır. Her yöntem için hesaplanan veri kapasite değerlerinin ortalamalarına ait grafik Şekil 4.4.'de verilmiştir. Şekil 4.4.'e göre veri gizleme kapasitesi en yüksek EM-4 ve EM-6 yöntemleri olmuştur. Literatürdeki Mielkainen ve Chan yöntemi dosyaya en fazla dosya boyutunun $\frac{1}{8}$'i oranında veri gizleyebildiği için maksimum 75KB (1bpp) veri gizleyebilmektedir. EM-4, ve EM-6 2-bitlik alana 4-bit gizlediği için maksimum 150KB (2bpp), EM-6 yöntemi 2-bitlik alana 3-bit gizlediği için 112,5KB (1,5bpp) literatürdeki yöntemlere göre daha fazla veri gizleyebilmektedir.



Şekil 4.4. 150 imgeye önerilen yöntemler ve literatür çalışmalarıyla farklı oranlarda veri gizlemeleriyle oluşan sırlı imgeler için gizleme oranları ortalamalarının karşılaştırılması

Geliştirilen sırörtme yöntemlerinin örtü imgede yaptığı değişiklikleri ölçebilmek ve literatürdeki eşdeğer iki çalışma ile her gizleme sonrasında oluşan sırlı imgenin değişen bit sayıları ölçülmüştür. Her yöntem için ölçülen değişen bit sayılarının ortalamalarına ailt grafik Şekil 4.5.'de verilmiştir. Şekil 4.5.'e göre önerilen 3 yöntemden tümü literatürdeki iki çalışmadan örtü imgeye aynı veriyi en az değişiklikle gizleyebilmiştir. Buna göre EM-4 ve EM-6 yöntemi ile 1bpp (75KB) veri gizleme durumunda yaklaşık 150.000 bit değişirken, EM-5 yöntemi 200.000 bit, Chan yöntemi 210.000 bit ve Mielkainen yöntemi 225.000 bit değiştirmektedir. Bu verilere göre EM-4 ve EM-6 yöntemi Mielkainen yöntemine göre yaklaşık 75.000 bit, Chan yöntemine göre ise 60.000 bit daha az değişiklikle aynı miktarda veriyi gizleyebilmiştir.



Şekil 4.5. 150 imgeye önerilen yöntemler ve literatür çalışmalarıyla farklı oranlarda veri gizlemeleriyle oluşan sırlı imgeler için değişen bit sayıları ortalamalarının karşılaştırılması

4.6. Önerilen Yöntemlerin Saldırılara Karşı Dayanıklılık Analizi

Tez çalışmasında geliştirilen yöntemlerin sıraçma ve atak yöntemlerine karşı dayanıklılıklarını Simple Steganalysis Suite, XStegsecret analiz yazılımları kullanılarak test edildi. Literatürdeki yöntemlerede aynı saldırılar yapılarak önerilen yöntemlerle karşılaştırıldı.

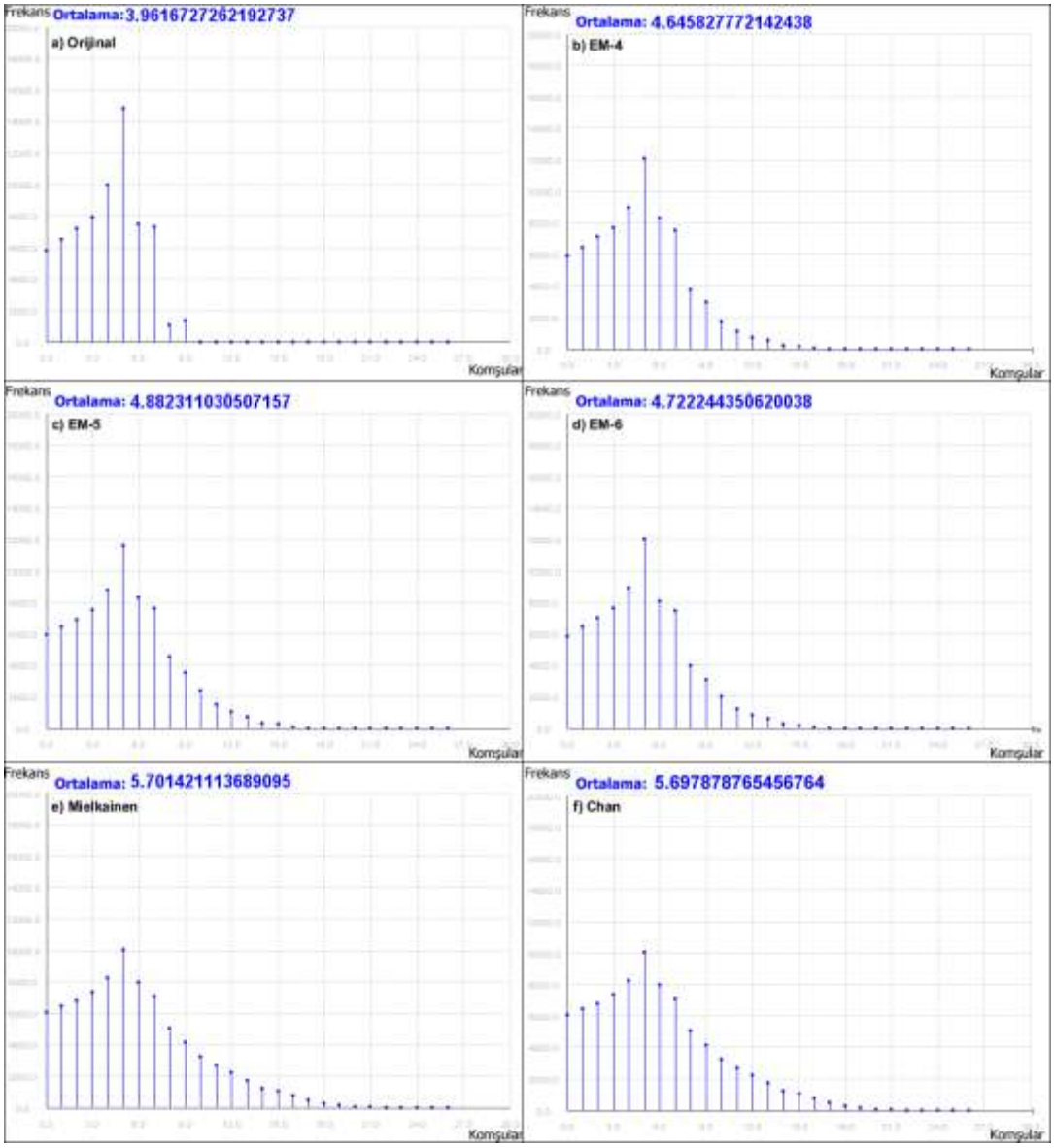
4.6.1. Sıraçma analizleri

Sıraçma yöntemleri olarak komşuluk histogramı, piksel farkı histogramı ve ki-kare analizi uygulanmıştır.

4.6.1.1. Komşuluk histogramı sıraçma yöntemi analizi

Geliştirilen sırörtme yöntemlerinin sıraçma yöntemlerine karşı dayanıklılığını ölçebilmek ve literatürdeki eşdeğer iki çalışma ile kıyaslayabilmek için 640x480 boyutlarındaki Lena imgesine tez çalışmasında önerilen 3 yöntem ile ve literatürdeki

iki yöntem ile 1bpp oranında veri gizlenmiştir. Her yöntemin oluşturduğu sırlı imgeler komşuluk (neighbourhood) histogramı sıraçma yöntemi ile analiz edilmiştir.



Şekil 4.6. 640x480 Lena imgesine önerilen yöntemler ve literatür çalışmalarıyla 1bpp veri gizlemeleriyle oluşan sırlı imgelerin komşuluk (neighbourhood) histogram sıraçma sonuçları a) Orjinal imge b) EM-4 ile veri gizlenmiş sırlı imge c) EM-5 d) EM-6 e) Mielika

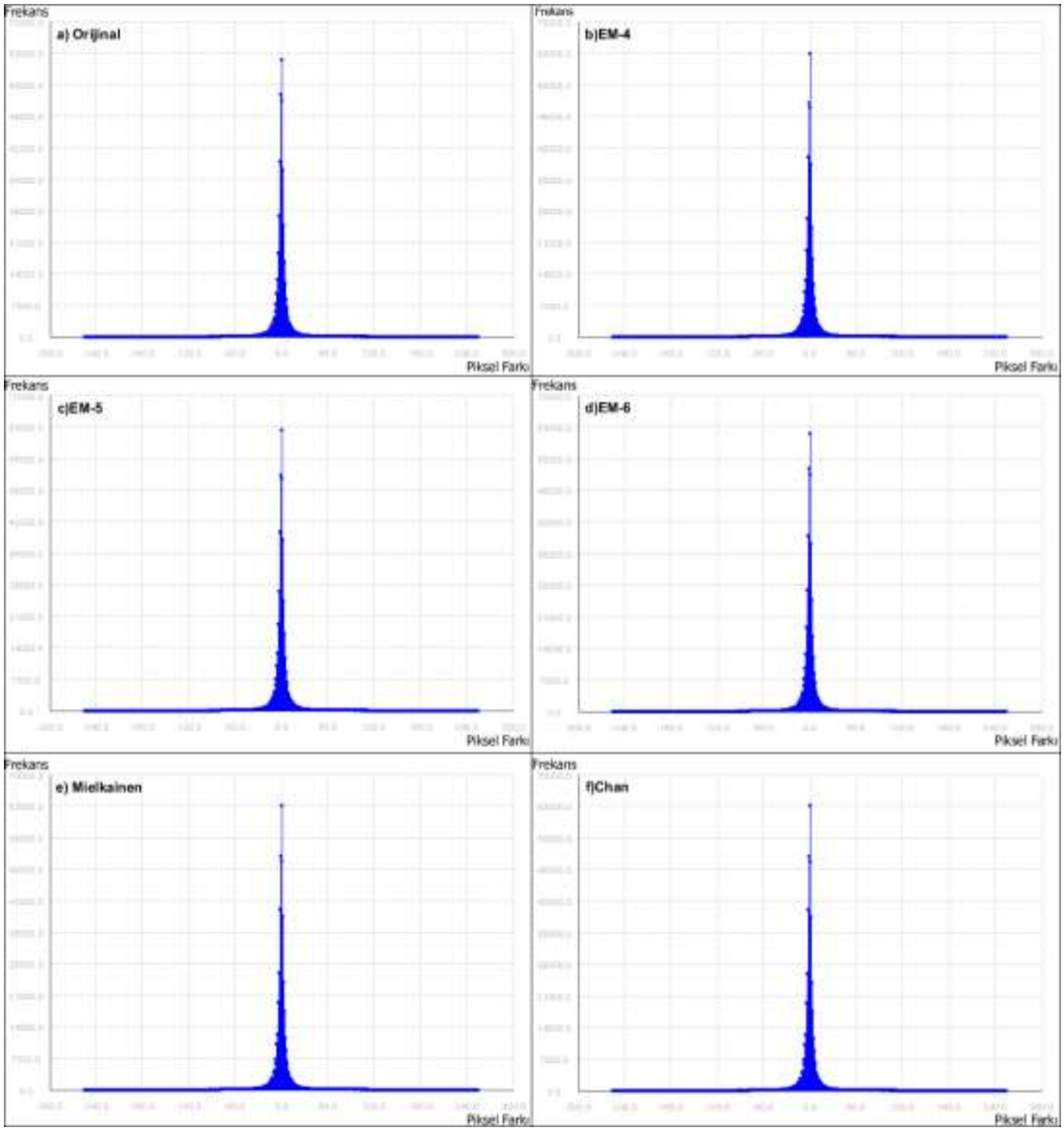
Analiz sonuçları Şekil 4.6.'da verilmiştir. Şekil 4.6.a ve b.'ye göre EM-4 yönteminin komşuluk histogramı analizi orjinal imgenin komşuluk histogramı analizine en yakın sonuç verdiği görülmektedir. Şekil 4.6.a.'da orjinal imgenin komşuluk histogram ortalaması 3,96 çıkarken Şekil 4.6.b.'de EM-4 yöntemiyle 1bpp veri gizlenmiş sırlı imgenin komşuluk histogram ortalaması 4,64 çıkmıştır. Diğer yöntemlerin komşuluk

histogram ortalamalarına bakarsak EM-5 yönteminin 4,88, EM-6 4,72, Mielkainen 5,70 ve Chan yönteminin 5,69 çıkmıştır. Buna göre komşuluk histogram analizine göre orjinal imgeye en benzerden en az benzere sırlı imge sıralaması EM-4, EM-6, EM-5, Chan ve Mielkainen şeklinde olmaktadır. Bu değerlerden de anlaşılacağı üzere EM-4 ve EM-5 ve EM-6 yöntemleri literatürdeki iki yönteme göre daha başarılı sonuçlar vermiştir. Bu sonuçlara dayanarak EM-4, EM-5 ve EM-6 yöntemlerinin örtü imgede daha az değişiklik yaptıkları görülmektedir.

4.6.1.2. Piksel farkı histogramı sıraçma yöntemi analizi

Komşuluk histogramında yapılan sıraçma analizininde kullanılan imgeler ile piksel farkı histogramı sıraçma yönteminde de kullanılmıştır. Her yöntemin oluşturduğu sırlı imgeler piksel farkı histogramı sıraçma yöntemi ile analiz edilmiştir. Analiz sonuçları Şekil 4.7.'de verilmiştir. Şekil 4.7.'deki tüm grafiklerde bütün yöntemler birbirine yakın sonuçlar vermiş gibi görünmektedir. Grafikler detaylıca incelendiğinde Şekil 4.7.a.'da orjinal imgenin piksel farkı histogramının tepe noktası 63.000 değerinin çok az altına ulaşmaktadır.

Şekil 4.7.b.'deki EM-4 yöntemiyle 1bpp veri gizlenmiş sırlı imgenin piksel farkı histogramının tepe noktası 63.000 değerinine ulaşmıştır. Şekil 4.8.'deki tüm grafikler incelendiğinde orjinal imgenin piksel farkı histogramına en yakın sonuç veren yöntemler EM-6 ve EM-5 yöntemi olmuştur.

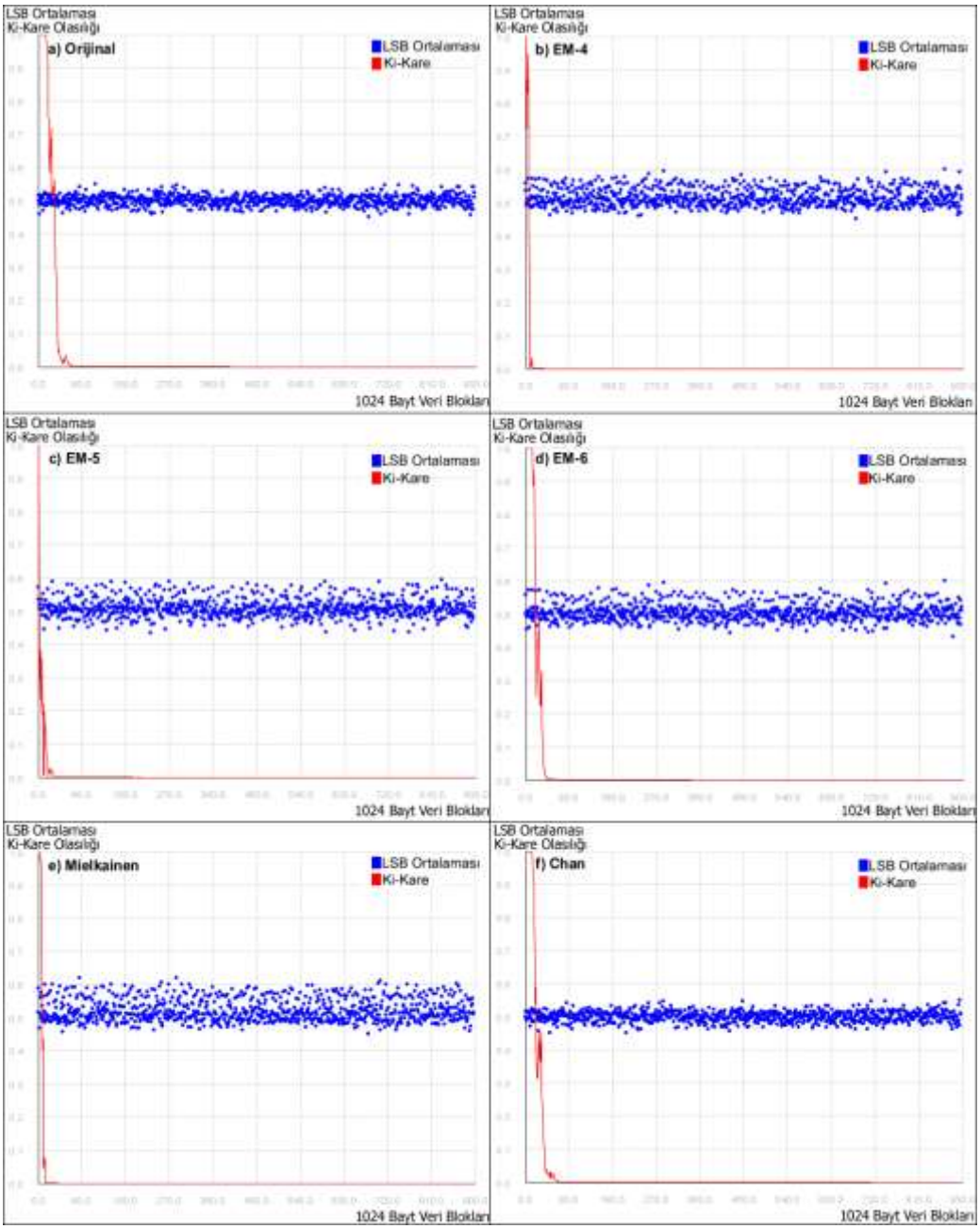


Şekil 4.7. 640x480 Lena imgesine önerilen yöntemler ve literatür çalışmalarıyla 1bpp veri gizlemeleriyle oluşan sırlı imgelerin piksel farkı histogram sıraçma sonuçları a) Orjinal imge b) EM-4 ile veri gizlenmiş sırlı imge c) EM-5 d) EM-6 e) Mielikainen [2] f) Chan[3]

Şekil 4.7.c,d,e,f grafikleri incelendiğinde EM-5, EM-6 yöntemlerinin piksel farkı histogramlarının tepe noktaları ile 40.000 değerinin az altında orijinal imge sonucuna en yakın değerleri üretmiştir. EM-4, Mielkainen ve Chan yönteminin histogramının tepe noktası tam 40.000 değerinin üzerindedir. Sonuç olarak piksel farkı histogramı sıraçma yönteminde EM-5 ve EM-6 yöntemleri en başarılı yöntemler olmuştur.

4.6.1.3. Ki-kare sıraçma yöntemi ve LSB dağılım analizi

Komşuluk histogramı ve piksel farkı histogramı sıraçma analizlerinde kullanılan imgeler, kikare sıraçma yönteminde de kullanılmıştır. Her yöntemin oluşturduğu sırlı imgeler kikare sıraçma yöntemi ile analiz edilmiştir. Analiz sonuçları Şekil 4.8.'de verilmiştir.



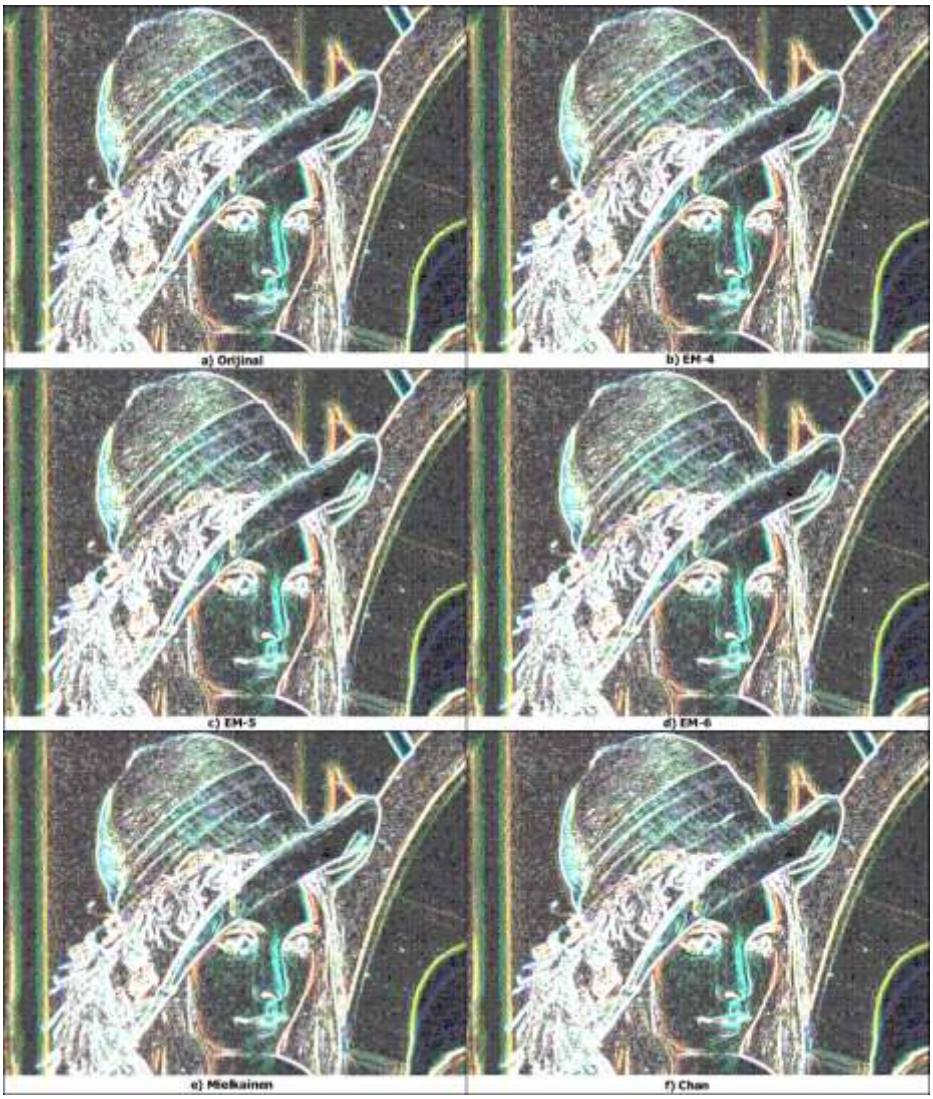
Şekil 4.8. 640x480 Lena imgesine önerilen yöntemler ve literatür çalışmalarıyla 1bpp veri gizlemeleriyle oluşan sırlı imgelerin ki-kare sıraçma sonuçları a) orjinal imgenin ki-kare sonucu b) EM-4 ile veri gizlenmiş sırlı imgenin ki-kare sonucu c) EM-5 d) EM-6 e) Mielkainen [2] f) Chan [3]

4.6.2. Atak analizleri

Piksel, LSB, salt & pepper, gaussian, speckle, poisson ve adjust color saldırı atakları kullanılarak önerilen yöntemler ile literatürdeki yöntemler kıyaslanmıştır.

4.6.2.1. Piksel atağı analizi

Sıraçma analizlerinde kullanılan imgelere aynı oranda piksel atağı yapılmıştır. Piksel atağı sonucunda oluşan imgeler Şekil 4.9.'da verilmiştir.



Şekil 4.9. 640x480 Lena imgesine önerilen yöntemler ve literatür çalışmalarıyla 1bpp veri gizlemeleriyle oluşan sırlı imgelerin piksel atağı sonucunda oluşan imgeler a) Orjinal imgenin piksel atağı sonucu b) EM-4 ile veri gizlenmiş sırlı imgenin piksel atağı sonucu c) EM-5 d) EM-6 e) Mielikainen [2] f) Chan [3]

Görsel analiz bölümünde de tüm gizleme yöntemlerinin örtü imgede gözle görülür bozulma yapmadığı gösterilmişti. Aynı şekilde her gizleme yöntemi ile veri gizlenmiş sırlı imgelere aynı oranda piksel atağı yapıldığında orijinal imgenin verdiği sonuçlar ile tüm yöntemlerin aynı sonucu verdiği gözlenmiştir. Piksel atağı sonucunda tüm yöntemlerde gözle görülür anormal bir bozukluğa rastlanmamıştır. Bu sonuç görsel analizi doğrular nitelikte olup yöntemlerin piksel atağı görsel tespit analizini geçtiğini göstermektedir.

4.6.2.2. En önemsiz bit atağı analizi

Sıraçma analizlerinde kullanılan imgelere en önemsiz bit kullanılarak atak yapılmıştır. Piksel atağı sonucunda oluşan imgeler Şekil 4.11.'de verilmiştir. Görsel analiz bölümünde ve piksel atağı analizinde de tüm gizleme yöntemlerinin örtü imgede gözle görülür bozulma yapmadığı gösterilmişti. Aynı şekilde her gizleme yöntemi ile veri gizlenmiş sırlı imgelere LSB atağı yapıldığında o tüm yöntemlerin aynı sonucu verdiği gözlenmiştir. LSB atağı sonucunda tüm yöntemlerde gözle görülür anormal bir bozukluğa rastlanmamıştır. Bu sonuç görsel analizi ve piksel atağı analizini doğrular nitelikte olup yöntemlerin LSB atağı görsel tespit analizini geçtiğini göstermektedir.

Şekil 4.10. 640x480 Lena imgesine önerilen yöntemler ve literatür çalışmalarıyla 1bpp veri gizlemeleriyle oluşan sırlı imgelere en önemsiz bit LSB atağı sonucunda oluşan imgeler a) Orjinal imgenin atak sonucu b) EM-4 ile veri gizlenmiş c) EM-5 d) EM-6 e) Mielikainen [2] f) Chan [3]

4.6.2.3. Parametrik atak analizi

Önerilen yöntemlerin ataklara karşı dayanıklılığı yöntemlerin başarımı açısından çok önemlidir. EM-4,EM-5 ve EM-6 yöntemleri ve literatürdeki iki yöntem salt & pepper, gaussian, poisson, speckle ve adjust color ataklarına karşı dayanıklılıkları analiz edilmiştir. Buna göre 640x480 boyutlarındaki Lena imgesine Şekil 4.11.'deki

Sakarya Üniversitesi logo imgesi beş yöntem ile gizlenmiştir. Daha sonra sırlı Lena imgelerine beş atak yöntemi ile saldırı yapılmış ve gizli logo imgesi tüm sırlı imgelerden geri çıkartılmıştır. Saldırı yapılmış sırlı imgelerden geri çıkarılan saü logoları orijinali ile kıyaslanarak imgedeki saldırı sonrasındaki değişim incelenmiştir. Başarım sonuçlarını içiren PSNR değerleri Tablo 4.5.'de verilmiştir.

Tablo 4.5. Önerilen ve literatürdeki yöntemlerin ataklara karşı başarım analizi kıyaslaması PSNR değerleri

Yöntem	salt&pepper	gaussian	poisson	speckle	adjust color
EM-6	24,12	8,37	8,31	8,43	5,42
EM-4	23,02	6.72	6,64	6,76	3,32
EM-5	21,6	5,47	5,67	5,91	2,58
Mielkainen	19,9	7,46	7,4	7,54	5,40
Chan	18,7	7,34	7,1	7,3	5,1

Önerilen yöntemler içerisinde saldırı ataklarına karşı en başarılı yöntem EM-6 olmuştur. Daha sonra sırayla EM-4 ve EM-5 gelmektedir. Özellikle salt & pepper tuz ve karabiber saldırısında üç yöntemde başarılı sonuçlar elde ederken diğer saldırı ataklarında daha fazla etkilenmişlerdir. En fazla etkilendikleri saldırı adjust color saldırısı olmuştur. En az ise salt & pepper atağından etkilenmişlerdir. Buna göre en az etkilenmeden en fazla etkilenmeye göre saldırılar sıralanırsa; salt & pepper, speckle, poisson, gaussian ve adjust color olmuştur. Mielkainen ve Chan birbirlerine yakın performanslar verseler de Mielkainen Chan'e göre daha başarılıdır. Önerilen yöntemler ile literatürdeki çalışmalar kıyaslandığında en başarılı yöntemin EM-6 yöntemi olduğu görülmüştür. Salt & pepper atağında önerilen yöntemlerin hepsi literatürdeki yöntemlere göre daha başarılıdır. Diğer ataklarda ise başarı sıralaması EM-6, Mielkainen, Chan, EM-4 ve EM-5 şeklinde olmuştur. Tablo 4.5.'deki sonuçlardan da görüldüğü üzere tez çalışmasında önerilen yöntemlerden EM-6 yöntemi saldırı ataklarına karşı literatürdeki yöntemlerden daha başarılıdır.

Şekil 4.11. 113x135 boyutlarındaki Sakarya Üniversitesi logo imgesi

4.7. Önerilen Yöntemlerin Histogram Ve Kalite Ölçütü Analizi

Tez çalışmasında önerilen yöntemlerden en başarılı veri gizleme yönteminin yapılan analizlerin sonucunda EM-4 olduğu tespit edilmiştir. Literatürdeki çalışmalardan tez çalışmasında önerilen imge içine kayıplı imge gizleyen EM-4 yöntemine en yakın yöntemlerden biri Jain ve Kumar'ın önerdiği yöntemdir [17]. Jain ve Kumar kayıplı veri sıkıştırması ile imge içine verimli veri gizleme yöntemini önermişlerdir. Jain ve Kumar yaptıkları çalışmalarında gizli veriyi kayıplı veri sıkıştırmasıyla gizlemişlerdir. EM-4 yöntemi ile Jain ve Kumar'ın yaptığı çalışmayı kıyaslayabilmek için histogram ve imge kalite ölçütleri kullanılmıştır. Jain ve Kumar önerdikleri yöntemde, tez çalışmasında önerilen yöntemler gibi renkli imgelere veri gizleyebilmektedir. Literatürdeki birçok yöntem gri-seviye imgelere veri gizlemek için tasarlanmıştır.

Şekil 4.12.a.'da gösterilen 313x289 boyutlarındaki 24-bit renkli orijinal Lena imgesine hem EM-4 yöntemiyle hemde Jain ve Kumar'ın önerdikleri yöntemle 1785-bayt veri gizlenmiştir. Sırlı imgeler histogram ve imge kalite ölçütleri ile analiz edilmiştir.

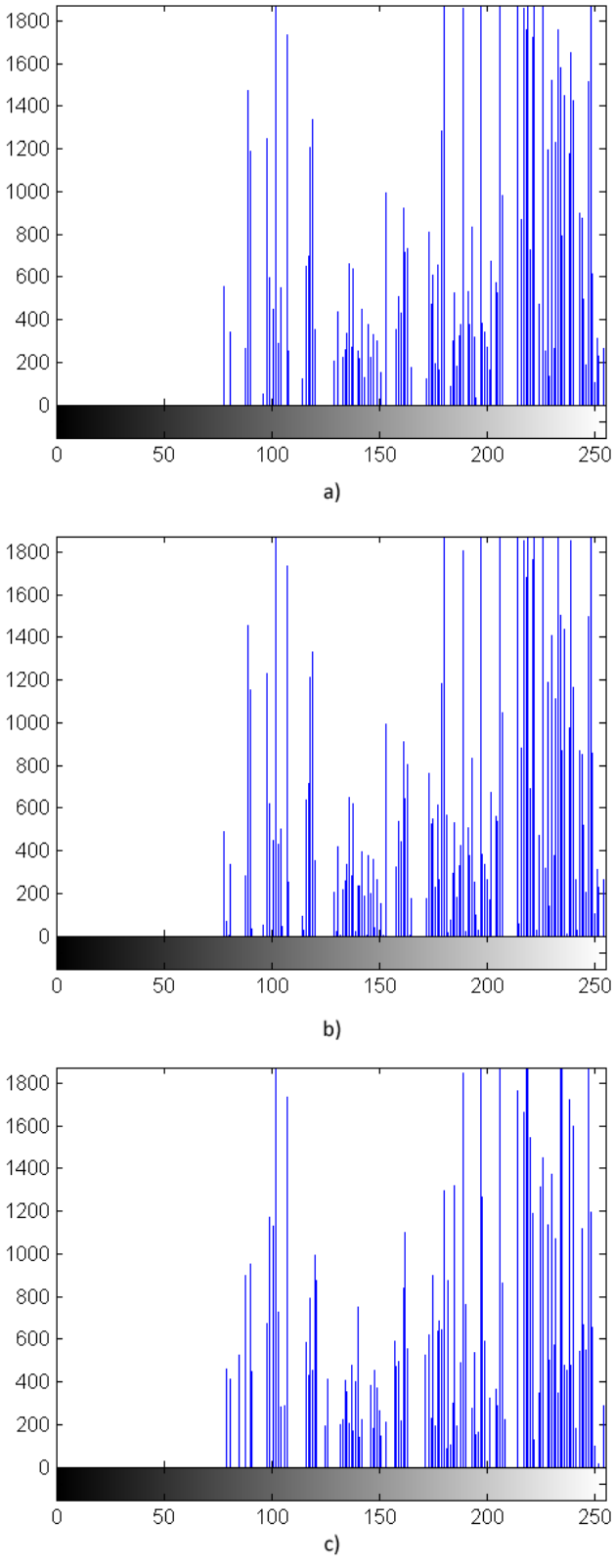
Şekil 4.13.'de orijinal Lena imgesinin, EM-4 yöntemi ve Jain ve Kumar'ın yöntemiyle 1785-bayt veri gizlenmiş sırlı imgelerin histogramları verilmiştir. Histogramlar incelendiğinde Şekil 4.13.b.'de EM-4 yönteminin veri gizlediği sırlı Lena imgesinin histogramı, Şekil 4.13.c.'deki Jain ve Kumar'ın yönteminin veri gizlediği sırlı Lena imgesinin histogramına göre orijinal Lena imgesinin histogramına daha yakındır. Buda Jain ve Kumar'ın önerdikleri yönteme göre örtü imgede daha az değişiklikle veri gizlediğini kanıtlamaktadır.

a)

b)

c)

Şekil 4.12. 313x289 24-bit Lena imgeleri a) orijinal Lena imgesi b) EM-4 yöntemiyle 1785-bayt veri gizlenmiş sırlı Lena imgesi c) Jain ve Kumar'ın önerdikleri yöntem [17] ile 1785-bayt veri gizlenmiş sırlı imge



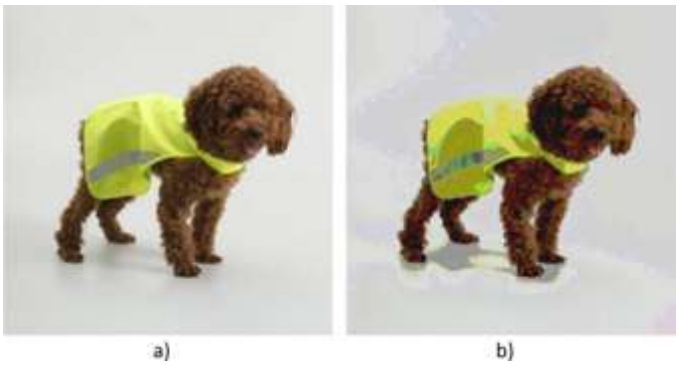
Şekil 4.13. 1785-bayt veri gizlenmiş Lena imgesinin histogramları a) orijinal lena imgesi b) EM-4 ile veri gizlenen sırlı Lena imgesi c) Jain ve Kumar'ın yöntemi [17] ile veri gizlenen sırlı Lena imgesi

Tablo 4.6.'da EM yöntemi ve Jain Kumar'ın önerdikleri yöntem ile 1785-bayt veri gizlenmiş sırlı imgelerin kalite ölçütleri ile analiz sonuçları verilmiştir. SSIM imge kalite ölçütüne göre EM-4 yönteminin veri gizlediği sırlı Lena imgesi 0,999 değeriyle neredeyse orijinal Lena imgesine %100 benzer olduğunu göstermektedir. UQI imge ölçütü, 1 değeri ile EM-4 yönteminin veri gizlediği sırlı Lenan imgesinin %100 oranında orijinal Lena imgesine benzediği sonucunu vermiştir. HPQM imge kalite ölçütüne göre ise Jain ve Kumar'ın yöntemi EM-4 yöntemine göre daha başarılı sonuç vermiştir. CQM ve PSNR değerlerine göre EM-4 yöntemi açık ara Jain ve Kumar'ın yöntemine göre daha başarılı sonuç vermiştir. Tablo verilerine göre EM-4 yöntemi, örtü imgede Jain ve Kumar'ın yöntemine göre daha az değişiklik yapmıştır.

Tablo 4.6. EM-4 yöntemi ve Jain ve Kumar'ın önerdiği yöntemin imge kalite ölçütleri başarımı

Yöntem	SSIM	UQI	HPQM	CQM	PSNR
EM-4	0,9999	1	0,8292	75,92	86,65
Jain ve Kumar [17]	0,9206	0,99	0,8405	43,45	33,73

Şekil 4.14.'de EM-4 yöntemi ile gizlenmiş orijinal köpek imgesi ve EM-4 yöntemiyle kayıplı olarak geri çıkarılmış köpek imgesi verilmiştir. EM-4 yöntemi kayıplı veri gizlediği için geri çıkarılan imgede bozulma oluşmuştur. Geri çıkarma işleminde imgede anlaşılmayacak kadar bozulma olmamış ama görsel olarak bazı değişiklikler oluşmuştur. İki imgenin benzerliklerini belirleyebilmek için kalite ölçütleri kullanılmış ve Tablo 4.7.'de imge kalite ölçütü sonuçları verilmiştir.



Şekil 4.14. Köpek imgeleri a) EM-4 yöntemi ile gizlenmiş orijinal köpek imgesi b) EM-4 yöntemi ile kayıplı geri çıkarılmış köpek imgesi

Tablo 4.7. EM-4 yöntemi ile gizlenmiş ve geri çıkarılmış köpek imgesinin imge kalite ölçütleri sonuçları

SSIM	UQI	HPQM	CQM	PSNR (db)
0,864	0,993	0,580	36,12	25,29

Tablo 4.7.'ye göre gizlenen köpek imgesi ile çıkarılan köpek imgesi SSIM kalite ölçütüne göre %86, UQI imge kalite ölçütüne göre %99 ve HPQM kalite ölçütüne göre ise %58 oranında benzeşmektedir. CQM kalite ölçütü 36,12 ve PSNR değeri ise 25,29 db olmuştur. Sonuçlara göre iki imge arasında benzerlik oranı yüksektir. Bu sonuçlara göre tez çalışmasında önerilen EM-4 yöntemi gizlenen imgeyi karşı tarafın anlayamacağı şekilde bozmamaktadır.

Bu bölümde, tez çalışmasında önerilen 6 sırörtme yönteminin imge kalite ölçütleri analizi, veri gizleme kapasitesi, örtü imgede yaptıkları değişiklik miktarı, sıraçma ve saldırı yöntemlerine göre dayanıklılıkları analiz edilmiştir. Yapılan bu analizler literatürde eşdeğer iyi bilinen iki yöntem için de yapılmıştır. Önerilen yöntemlerin başarım performansları literatürdeki eşdeğer çalışmaların sonuçları ile kıyaslanmıştır. Ayrıca tez çalışmasıyla literatürdeki benzer çalışma histogram ve imge kalite ölçütleri ile kıyaslanarak sonuçları verilmiştir. Tez çalışmasında önerilen yöntemlerin birçok kıstasta daha başarılı sonuçlar verdiği gözlemlenmiştir.

Bir sonraki bölümde tez çalışmasının sonucu ve gelecekte önerilen yöntemler ile ilgili yapılacaklar verilecektir.

BÖLÜM 5. SONUÇLAR

İnsanların ve makinelerin birbirleri ile yoğun bir şekilde haberleştiği çağımızda güvenli iletişim konusu önemini her geçen gün arttırarak korumaktadır. Tez çalışmasında güvenli iletişim tekniklerinden olan sırörtme yöntemlerine bir yenisinin eklenerek gerçeklenmesi amaçlanmıştır. Tez çalışmasında geliştirilen yeni sırörtme yöntemi ve bu sırörtme yönteminden türetilmiş beş türev yöntem geliştirilmiştir. Geliştirilen veri gizleme yöntemi sayısal imgelere insan göz sistemi tarafından algılanamayacak şekilde verileri gizler. Geliştirilen yöntemlerde en temel amaç veri gizlenen örtü imgeyi en az oranda değiştirerek veriyi gizlemektir. Tez çalışmasında önerilen gizleme yöntemleri literatüre üç katkı sağlamıştır.

1. Geliştirilen yöntemler literatürdeki çalışmalara göre örtü imgede daha az değişiklik yapmaktadır.
2. Literatürdeki çoğu yöntem 2-bitlik veri gizlemesinde yaklaşık 1-bit değiştirirken, EM-4 yöntemi 4-bit veri gizlemesinde 1-bit değişiklik oluşturmaktadır. EM-5 yöntemi 3-bitlik veri gizlemesinde 1-bit, EM-6 yöntemi ise 2-bitlik veri gizlemesinde 1-bit değişiklik oluşturmaktadır.
3. Geliştirilen yöntemler literatürdeki birçok yöntem gibi gizli veriyi son bitlere gizlememekte bunun yerine her bloğa gizlenecek veri parçasının aralığı gizlemektedir. Son bitlerden elde edilen veri kod sistemini geriye çıkardığı için gizli veriye doğrudan erişilemez. Gizli verinin çözümü için kod sisteminin bilinmesi gereklidir. Bu özellikte geliştirilen yöntemleri sıraçma ataklarına karşı ekstra güvenli yapmaktadır.
4. EM-4 yöntemi literatürdeki çalışmalara göre aynı veriyi daha az değişiklikle gizlerken aynı zamanda literatürdeki çalışmalara göre 2 kat daha fazla veri gizleyebilmektedir.

Tez çalışmasında geliştirilen yöntemlerden EM-1, EM-2 ve EM-3 yöntemleri veri çıkarma işlemi için orijinal imgeye ihtiyaç duymaktadır. Bu zayıflığı ortadan kaldırmak için tasarlanan EM-4, EM5 ve EM-6 yöntemleri veri çıkarma işleminde orjinal imgeye ihtiyaç duymamaktadır.

EM-1, EM-2 ve EM-3 yöntemlerindeki diğer bir zayıflık ise gizlenecek veri parçasının bloktaki eşleştirme alanları ile eşleşmeme ihtimalidir. Bu durum önerilen yöntemlerin veri kapasitesini düşürmektedir. Özellikle karmaşık görüntüler içeren imgelerde eşleştirme alanı bilgileri birbirinden farklı olarak çeşitlenmekte ve eşleşme oranı daha yüksek çıkmaktadır. Fakat düz renkler içeren imgelerde eşleşme alanı bilgilerinin çeşitliliğin azalması eşleşme oranını düşürmektedir. Bu zayıflığı ortadan kaldırmak için tasarlanan EM-4, EM5 ve EM-6 yöntemlerinde eşleşme alanına eşleşmeme gibi bir durum söz konusu değildir. Buna karşın yaklaşık eşleştirme mantığı ile gizlenecek veriler hiçbir blok atlamadan sırayla gizlenir. Ayrıca veri gizleme kapasiteleri ilk geliştirilen yöntemlere göre yaklaşık EM-4 yönteminde 16 kata kadar artmıştır. Bu kadar büyük arttışın nedeni ilk yöntemlerde eşleşmeyen blokların kullanılmaması ve oldukça eşleşmeyen blok olmasıdır. Böylece veri gizleme kapasitesi problemine de çözüm getirilmiştir.

Tezin devamı niteliğinde yeni çalışma alanları ve konuları olarak yaklaşık eşleştirme yöntemleri için gizlenen imgenin piksellerindeki renk tonlarına göre veri çıkarma aşamasında yüksek, orta ve düşük aralıklarından en iyi eşleşen seviyeyi bularak çıkarılan imgenin kalitesini yükseltmek için iyileştirmeler yapılması planlanmaktadır. Ayrıca, sırörtme yöntemlerini daha nesnel açıdan değerlendirme imkânı verecek, MSE, PSNR ve kapasite paremetrelerinden elde edilecek yeni bir metrik geliştirilerektir.

KAYNAKLAR

[1] T. Sharp, "An Implementation of Key-Based Digital Signal Steganography," in *International Workshop on Information Hiding*, 2001, pp. 13–26.

[2] J. Mielikainen, "LSB matching revisited," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 285–287, 2006.

[3] C. Chan, "On Using LSB Matching Function for Data Hiding in Pixels," vol. 96, pp. 49–59, 2009.

[4] J. Tian, "Reversible Data Embedding Using a Difference Expansion," *IEEE Trans. Circuits Syst.*, vol. 13, no. 8, pp. 890–896, 2003.

[5] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Trans. Image Process.*, vol. 13, no. 8, pp. 1147–1156, 2004.

[6] C. C. Chang, Y. C. Chou, and T. D. Kieu, "Information hiding in dual images with reversibility," *3rd Int. Conf. Multimed. Ubiquitous Eng. MUE 2009*, pp. 145–152, 2009.

[7] T.-C. Lu, C.-Y. Tseng, and J.-H. Wu, "Dual imaging-based reversible hiding technique using LSB matching," *Signal Processing*, vol. 108, pp. 77–89, Mar. 2015.

[8] A. D. Ker, "Quantitive evaluation of pairs and RS steganalysis," in *in Security, Steganography, and Watermarking of Multimedia Contents*, 2004, pp. 89–97.

[9] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1613–1626, Jun. 2003.

[10] C. M. Wang, N. I. Wu, C. S. Tsai, and M. S. Hwang, "A high quality steganographic method with pixel-value differencing and modulus function," *J. Syst. Softw.*, vol. 81, no. 1, pp. 150–158, 2008.

[11] J. Fridrich and D. Soukal, "Matrix Embedding for Large Payloads," *IEEE Trans. Inf. Forensics Secur.*, vol. 1, no. 3, pp. 390–395, Sep. 2006.

[12] O. Kurtuldu and N. Arica, "A new steganography method using image layers," *2008 23rd Int. Symp. Comput. Inf. Sci.*, pp. 1–4, 2008.

[13] M. Soleimanpour-Moghadam and H. Nezamabadi-pour, "The pair-wise LSB matching steganography with a discrete quantum behaved Gravitational Search Algorithm," *J. Intell. Fuzzy Syst.*, vol. 30, no. 3, pp. 1547–1556, Mar. 2016.

[14] H. Wu, H. Wang, Y. Hu, and L. Zhou, "Digital-Forensics and Watermarking: 13th International Workshop, IWDW 2014, Taipei, Taiwan, October 1-4, 2014. Revised Selected Papers," Y.-Q. Shi, J. H. Kim, F. Pérez-González, and C.-N. Yang, Eds. Cham: Springer International Publishing, 2015, pp. 455–469.

[15] F. Huang, Y. Zhong, and J. Huang, "Improved algorithm of edge adaptive image steganography based on LSB matching revisited algorithm," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8389 LNCS, pp. 19–31.

[16] V. Sabeti, S. Samavi, and S. Shirani, "An adaptive LSB matching steganography based on octonary complexity measure," *Multimed. Tools Appl.*, vol. 64, no. 3, pp. 777–793, 2013.

[17] R. Jain and N. Kumar, "Efficient data hiding scheme using lossless data compression and image steganography," vol. 4, no. 08, pp. 3908–3915.

[18] C. Olcay, "İmge İçine Bilgi Gizlemede Kullanılan LSB Yontemlerinin Karşılaştırması," *Çankaya Univ. J. Sci. Eng.*, vol. 10, no. 1, pp. 17–32, 2013.

[19] A. H. Murray and R. W. Burchfiled, "The Oxford English Dictionary: Being Corrected Re-issue," 1993.

[20] A. Şahin, "New Methods on Image Steganography and Their Reliabilities," Trakya University, Natural and Applied Sciences, Department of Computer Enginering, 2007.

[21] Ö. Çetin, "Hareketli Görüntü Uygulamaları için Sırörtme Yaklaşımı ile Veri Gömme Algoritmasi Tasarımı," Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği, Doktora Tezi, 2008.

[22] Y. Yalman, "Sayısal Görüntüler İçin Histogram Temelli Veri Gizleme Yöntemi Ve Uygulama Yazılımı," Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği, Doktora Tezi, 2010.

[23] S. Emek, "Sabit Görüntüler Ve Video İşaretleri İçin Ayrık Dalgacık Dönüşümü Ayrık Kosinüs Dönüşümü Tabanlı Sayısal Damgalama Yöntemi," Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Haberleşme, Doktora Tezi, 2006.

[24] F. Akar and H. S. Varol, "A New RGB Weighted Encoding Technique for Efficient Information Hiding in Images," *Deniz Bilimleri ve Mühendisliği Dergisi*, vol. 2, no. 2. 2004.

[25] J. J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking digital images for copyright protection," *IEE Proc. - Vision, Image Signal Process.*, vol. 143, no. 4, p. 250, 1996.

[26] C. I. Podilchuk and W. Zeng, "Digital image watermarking using visual models," in *Electronic Imaging '97*, 1997, pp. 100–111.

[27] J. R. Hernández, M. Amado, and F. Pérez-González, "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 55–68, 2000.

[28] A. Durdu and A. T. Özcerit, "Sırörtülü Ses Dosyalarının Yapay Zeka Yöntemleri Yardımıyla Çözümlenmesi," Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Eğitimi, Yüksek Lisans Tezi, 2010.

[29] W. Bender, D. Gruhl, N. Morimoto, and a. Lu, "Techniques for data hiding," *IBM Syst. J.*, vol. 35, no. 3.4, pp. 313–336, 1996.

[30] M. H. Yavuz, "Müzikle Şifreleme-Veri Gizleme Sistemi Tasarımı ve Gerçeklenmesi," TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Ensitüsü, Bilgisayar Mühendisliği, Yüksek Lisans Tezi, 2010.

[31] F. Petitcolas, "MP3Stego." [Çevrimiçi]. Mevcut: http://www.petitcolas.net/steganography/mp3stego/index.html. [Erişim Tarihi: 10-Aralık-2015].

[32] K. Gopalan, "Audio steganography using bit modification," in *2003 International Conference on Multimedia and Expo. ICME '03. Proceedings (Cat. No.03TH8698)*, 2003, vol. 1, pp. I–629.

[33] J. Chou, K. Ramchandran, and A. Ortega, "High capacity audio data hiding for noisy channels," in *Proceedings International Conference on Information Technology: Coding and Computing*, 2001, pp. 108–112.

[34] O. Ünlü, "Ortam Ve Yöntem Bağımsız Steganografik Kütüphane Tasarımı," Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği, Yüksek Lisans Tezi, 2012.

[35] M. Shirali-Shahreza, "A New Method for Real-Time Steganography," *2006 8th Int. Conf. Signal Process.*, 2006.

[36] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "A New Approach to Persian/Arabic Text Steganography," in *5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering,Software Architecture and Reuse (ICIS-COMSAR'06)*, 2006, pp. 310–315.

[37] "Spammimic." [Çevrimiçi]. Mevcut: https://www.spammimic.com. [Erişim Tarihi: 10-Aralık-2015].

[38] T. Jamil, "Steganography: the art of hiding information in plain sight," *IEEE Potentials*, vol. 18, no. 1, pp. 10–12, 1999.

[39] Ç. Dereli, "Dilbilimsel Steganografi Yöntemleri Üzeriine Biir Araştırma," Ege Üniversitesi, Fen Bilimleri Enstitüsü, Uluslararası Bilgisayar Enstitüsü, Yüksek Lisans Tezi, 2010.

[40] H. Lashkari, A. Abdul Manaf, M. Masrom, and S. Mohd. Daud, "A survey on image steganography algorithms and evaluation." Springer Berlin Heidelberg, 07-Nov-2011.

[41] M. K. Al-karawi, "İkili Görüntü Kullanarak İmza Steganografi Modellerinin Geliştirilmesi," Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektronik Bilgisayar Eğitimi, Yüksek Lisans Tezi, 2012.

[42] Z. Erkin, B. Örencik, B. Mühendisliği, B. Elektrik-Elektronik, F. İstanbul, T. Üniversitesi, M. İstanbul, A. Sözcükler, Steganografi, G. Kriptografi, İ. Yazı, G. Güvenliği, and İletişim, "Steganografik Kütüphane."

[43] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, Mar. 2010.

[44] N. F. Johnson and S. C. Katzenbeisser, "A Survey of Steganographic Techniques," in *Information Hiding Techniques for Steganography anda Digital Watermarking*, pp. 43–78.

[45] K.-H. Jung and K.-Y. Yoo, "Data hiding method using image interpolation," *Comput. Stand. Interfaces*, vol. 31, no. 2, pp. 465–470, Feb. 2009.

[46] M. X. E. Chrysochos, V. Fotopoulos, A. N. Skodras, "Reversible Image Watermarking Based on Histogram Modification," in *11th Panhellenic Conference in Informatics*, 2007, pp. 93–104.

[47] H.-C. Huang and W.-C. Fang, "Intelligent Multimedia Data Hiding Techniques and Applications," in *2008 International Conference on Information Security and Assurance (isa 2008)*, 2008, pp. 477–482.

[48] G. Bhatnagar and B. Raman, "A new robust reference watermarking scheme based on DWT-SVD," *Comput. Stand. Interfaces*, vol. 31, no. 5, pp. 1002–1013, Sep. 2009.

[49] A. Gurijala, J. R. D. Jr, M. S. Seadle, and J. H. L. Hansen, "Speech watermarking through parametric modeling.," in *7th International Conference on Spoken Language Processing, ICSLP2002 - INTERSPEECH 2002, Denver, Colorado, USA, September 16-20, 2002*, 2002.

[50] M. Tunçkanat and Ş. Sağıroğlu, "A Secure Internet Communication Tool," *Turkish J. Telecommun.*, vol. 1, no. 1, pp. 40–46, 2002.

[51] Hsien-Wen Tseng and Chin-Chen Chang, "Steganography using JPEG-compressed images," in *The Fourth International Conference onComputer and Information Technology, 2004. CIT '04.*, 2004, pp. 12–17.

[52] G. Brisbane, R. Safavi-Naini, and P. Ogunbona, "High-capacity steganography using a shared colour palette," in *Vision, Image and Signal Processing, IEE Proceedings*, 2005, pp. 782–792.

[53] Y. K. Lee and L. H. Chen, "High capacity image steganographic model," *IEE Proc. - Vision, Image, Signal Process.*, vol. 147, no. 3, p. 288, 2000.

[54] R. Anderson, *Information Hiding*, vol. 1174. Berlin, Heidelberg: Lecture Notes in Computer Science,Springer Berlin Heidelberg, 1996.

[55] A. U. Yargıçoğlu, "Düşük Veri Hızlarında Çalışan Konuşma Kodlayıcılarına Gürbüz Bilgi Saklama ve Damgalama," Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektirik Elektronik Mühendisliği, Doktora Tezi, 2010, 2010.

[56] L. Chang and I. S. Moskowitz, "Critical analysis of security in voice hiding techniques," in *ICICS '97 Proceedings of the First International Conference on Information and Communication Security*, 1997, pp. 203–216.

[57] C. Kratzer, J. Dittmann, T. Vogel, and R. Hillert, "Design and evaluation of steganography for voice-over-IP," in *2006 IEEE International Symposium on Circuits and Systems*, p. 4.

[58] C. Xu, X. Ping, and T. Zhang, "Steganography in Compressed Video Stream," in *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, vol. 1, pp. 269–272.

[59] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283–301, 1998.

[60] M. D. Swanson, B. Chau, and A. H. Tewfik, "Multiresolution video watermarking using perceptual models and scene segmentation," in *Proceedings of International Conference on Image Processing*, vol. 2, pp. 558–561.

[61] O. Cetin and a. T. Ozcerit, "A new steganography algorithm based on color histograms for data embedding into raw video streams," *Comput. Secur.*, vol. 28, no. 7, pp. 670–682, Oct. 2009.

[62] T. Kalker, G. Depovere, J. Haitsma, and M. J. Maes, "Video watermarking system for broadcast monitoring," in *Electronic Imaging '99*, 1999, pp. 103–112.

[63] A. Durdu, "Test seti ve matlab kodları," 2016. [Çevrimiçi]. Mevcut: https://drive.google.com/a/sakarya.edu.tr/file/d/0B-Ku_tMBJJbSSHBfZkpNdW9wM3c/view?usp=sharing. [Erişim Tarihi: 01-Nisan-2016].

[64] Y. Yalman, "Histogram based perceptual quality assessment method for color images," *Comput. Stand. Interfaces*, vol. 36, no. 6, pp. 899–908, 2014.

[65] Y. Yalman and İ. ERTÜRK, "A new color image quality measure based on YUV transformation and PSNR for human vision system," *Turkish J. Electr. Eng.*, vol. 21, no. 2, pp. 603–612, 2013.

[66] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

[67] W. Zhou and a C. Bovik, "A universal image quality index," *Signal Process. Lett. IEEE*, vol. 9, no. 3, pp. 81–84, 2002.

ÖZGEÇMİŞ

1982 yılında Ankara'da doğdu. 1999 yılında Ankara Balgat Teknik Lisesi'nden mezun oldu. Daha sonra 2001 yılında Ankara Üniversitesi Çankırı Meslek Yüksekokulu Bilgisayar Programcılığı bölümünden birincilikle mezun oldu. Aynı yıl Ankara'da özel bir yazılım şirketinde yazılım geliştirici olarak çalıştı. 2003 yılında Sakarya Üniversitesi Bilgisayar Sistemleri Öğretmenliği bölümünü kazanarak tekrar eğitimine devam etti. 2007 yılında bölümünden birincilikle mezun olduktan sonra 2 yıl Sakarya Üniversitesi Bilgi İşlem Daire Başkanlığında web yazılım uzmanı olarak görev yaptı. Daha sonra Sakarya Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezine atanarak aynı görevine devam etti. 2008 yılında Sakarya Üniversitesi'nde Yüksek Lisans öğrenimi görmeye başladı. 2010 yılında Yüksek Lisans eğitimini tamamladı. 2010 yılında Bilgisayar Mühendisliği bölümünde Doktora eğitimine başladı. 2013 yılında öğretim görevlilliğine atandı. Bu görevine halen devam etmektedir. Ayşe Nehir ve Alya Hüma adında iki kızı vardır.