T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ MULTİMEDYA SENSÖR AĞLARDA (WMSN) GÜVENLİ VERİ İLETİŞİM İÇİN ALTERNATİF BİR UYGULAMA

Cebrail BARUT

Yüksek Lisans Tezi

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Donanım Bilim Dalı

OCAK 2020

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

KABLOSUZ MULTİMEDYA SENSÖR AĞLARDA (WMSN) GÜVENLİ VERİ İLETİŞİM İÇİN ALTERNATİF BİR UYGULAMA

Tez Yazarı
Cebrail BARUT

Danışman
Prof. Dr. Yetkin TATAR

OCAK 2020
ELAZIĞ

T. C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Başlığı: Kablosuz Multimedya Sensör Ağlarda (WMSN) Güvenli Veri İletişim İçin Alternatif Bir Uygulama

Yazarı: Cebrail BARUT

İlk Teslim Tarihi: 25.12.2019

Savunma Tarihi: 20.1.2020

TEZ ONAYI

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına göre hazırlanan bu tez aşağıda imzaları bulunan jüri üyeleri tarafından değerlendirilmiş ve akademik dinleyicilere açık yapılan savunma sonucunda OYBİRLİĞİ ile kabul edilmiştir.

İmza

Danışman: Prof. Dr. Yetkin TATAR — Onayladım
Fırat Üniversitesi, Fen Fakültesi

Başkan: Dr. Öğr. Üyesi Nuh ALPASLAN — Onayladım
Bingöl Üniversitesi, Mühendislik Fakültesi

Üye: Prof. Dr. Yetkin TATAR — Onayladım
Fırat Üniversitesi, Mühendislik Fakültesi

Üye: Dr. Öğr. Üyesi Güngör YILDIRIM — Onayladım
Fırat Üniversitesi, Mühendislik Fakültesi

Bu tez, Enstitü Yönetim Kurulunun ...../...../20..... tarihli toplantısında tescillenmiştir.

İmza

Prof. Dr. Soner ÖZGEN
Enstitü Müdürü

BEYAN

Fırat Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım "Kablosuz Multimedya Sensör Ağlarda (WMSN) Güvenli Veri İletişim İçin Alternatif Bir Uygulama" Başlıklı Yüksek Lisans Tezimin içindeki bütün bilgilerin doğru olduğunu, bilgilerin üretilmesi ve sunulmasında bilimsel etik kurallarına uygun davrandığımı, kullandığım bütün kaynakları atıf yaparak belirttiğimi, maddi ve manevi desteği olan tüm kurum/kuruluş ve kişileri belirttiğimi, burada sunduğum veri ve bilgileri unvan almak amacıyla daha önce hiçbir şekilde kullanmadığımı beyan ederim.

20/1/2020

Cebrail BARUT

ÖNSÖZ

Kablosuz multimedya algılayıcı ağlar son yıllarda popülerlik kazanmış olup; herhangi bir altyapı kurulumuna gerek duyulmaksızın ilgilenilen bir alandan ses ve görüntü gibi medya verilerini alıp bir merkeze gönderme kabiliyetine sahiptirler. Bu ağlarda kullanılan algılayıcı düğümler, sınırlı bir enerji kaynağına sahip olan bir bataryadan beslendiği için enerji verimliliği hayati önem taşımaktadır. Bu çalışmada kablosuz multimedya algılayıcı ağlarda kullanılmak üzere bir resim sıkıştırma algoritması geliştirilmiştir. Sıkıştırılan verinin yetkisiz kişiler tarafından ele geçirilmesini engellemek amacıyla sıkıştırma performansını göz önünden bulundurularak bir şifreleme işlemi gerçekleştirilmiştir.

Bu çalışmada değerli zamanını ayırıp çalışmamın tamamlanmasından her türlü yardımı yapan danışman hocam Prof. Dr. Yetkin TATAR'a teşekkür ederim.

Ayrıca çalışmalarım esnasında ihmal ettiğim aileme, eşime ve kızıma bana verdikleri destekten dolayı teşekkür ederim.

Cebrail BARUT
ELAZIĞ, 2020

İÇİNDEKİLER

Sayfa

ÖZET

Kablosuz Multimedya Sensör Ağlarda (WMSN) Güvenli Veri İletişim İçin Alternatif Bir Uygulama

Cebrail BARUT

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Ocak 2020, Sayfa: xii + 70

Kablosuz multimedya algılayıcı ağlar çok farklı ortamlarda (kara, deniz ve yeraltı) kullanılmaktadır. Herhangi bir altyapı kurulum maliyeti gerektirmeyen bu ağlarda kullanılan algılayıcı düğümlerin sınırlı bir enerji kaynağına sahip olması nedeniyle, sistem ömrünün uzun olması için enerjinin verimli bir şekilde kullanılması problemi ortaya çıkmıştır.

Medya verisi (ses veya görüntü) sıkıştırılabilir bir veri türüdür. Algılayıcı düğüm tarafından veri transferi aşamasında maksimum enerji tüketimi gerçekleştirilir. Bu bağlamda veri boyutunun azaltılması tüketilen enerji miktarını azaltmaktadır.

Bu çalışmada kablosuz multimedya algılayıcı ağlarda kullanılabilecek ve resim sıkıştırma performansını arttırmayı amaçlayan bir yöntem geliştirilmiştir. Yöntem algılayıcı düğümün daha önce çekmiş olduğu varsayılan bir referans resmi (arka plan) ile mevcut durumda çektiği resim (ön plan) arasında değişen alanlarda, değişim derecesine göre sıkıştırma işlemi gerçekleştirme esasına dayanmaktadır. Önerilen yöntem önemsiz olarak nitelenen alanda bir tek sınıflandırma yapmasına karşın; önemli olarak nitelenen alanda önem derecesine göre farklı sayıda sınıflandırma yaparak sıkıştırma işlemi gerçekleştirmektedir. Ayrıca sıkıştırılmış olan verinin yetkisiz kişiler tarafından ele geçirilmesini engellemek amacıyla iletilecek olan verinin boyutunda önemli bir artış olmadan bir şifreleme işlemi gerçekleştirilmiştir.

Anahtar Kelimeler: Kablosuz Multimedya Algılayıcı Ağlar, Resim Sıkıştırma, Resim Şifreleme

ABSTRACT

An alternative application for Secure Data Communication in wireless multimedia Sensor Networks (WMSN)

Cebrail BARUT

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Computer Engineering

January 2020, Pages: xii + 70

Wireless multimedia sensor networks are used in many different environments (land, sea and underground). The fact that the sensing nodes used in these networks, which do not require any infrastructure installation costs, have a limited energy source, has caused the problem of efficient use of energy for long system life. Media data (audio or video) is a compressible data type. Maximum energy consumption is realized by the sensor node in the data transfer stage. In this context, reducing the data size reduces the amount of energy consumed.

In this study, a method which can be used in wireless multimedia sensor networks and which aims to improve image compression performance has been developed. The method is based on performing compression according to the degree of variation in the areas varying between a presumed reference picture previously taken by the sensor node and the picture it is currently taking. With the proposed method, it performs compression process according to the importance level in the area which is considered important. In addition, an encryption process was performed without increasing the size of the data to be transmitted in order to prevent the unauthorized persons from capturing the compressed data.

Keywords: Wireless Multimedia Sensor Networks, Image Compression, Image Encryption

ŞEKİLLER LİSTESİ

Sayfa

TABLOLAR LİSTESİ

Sayfa

SİMGELER VE KISALTMALAR

Kısaltmalar

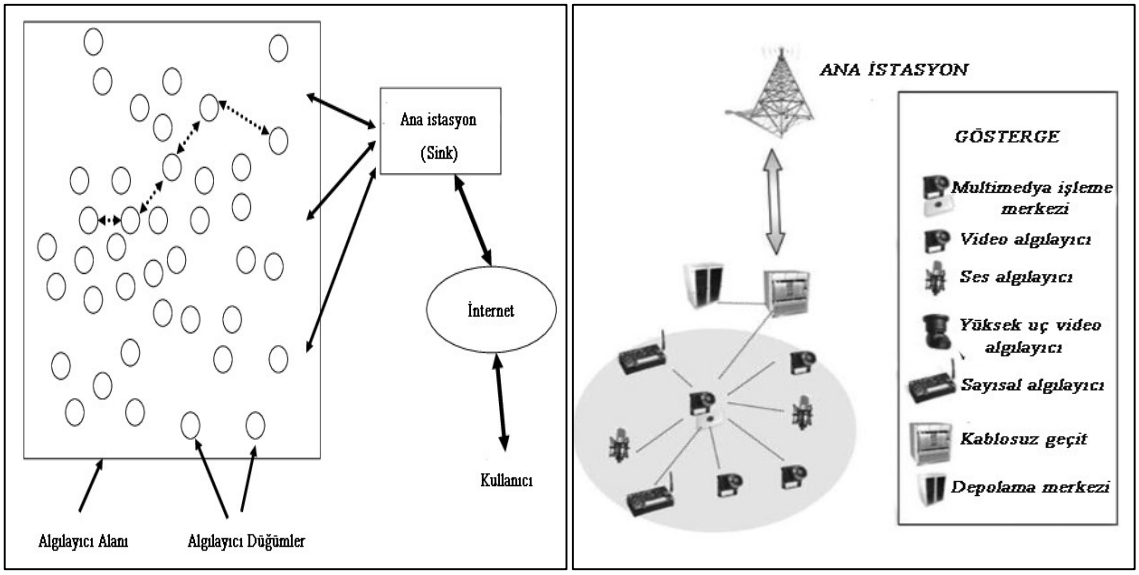
ADC : Analog Digital Converter
AES : Advanced Encryption Standard
AKD : Ayrık Kosinüs Dönüşümü
D-LEICA : Değişken-Low Energy Image Compression Algorithm
BWT : Burrows–Wheeler Transform
DCT : Discrete Cosine Transform
DES : Data Encryption Standard
DSA : Digital Signature Algorithm
EBCOT : Embedded Block Coding with Optimized Truncation
ECDSA : Elliptical Curve Digital Signature Algorithm
EZW : Embedded Zerotrees of Wavelet transforms
GIF : Graphics Interchange Format
IEEE : Institute of Electrical and Electronics Engineers
JPEG : Joint Photographic Experts Group
KAA : Kablosuz Algılayıcı Ağlar
KMA : Kablosuz Multimedya Algılayıcı
KMAA : Kablosuz Multimedya Algılayıcı Ağlar
LEICA : Low Energy Image Compression Algorithm
LZW : Lempel–Ziv–Welch
MAC : Medium Access Control
mJ :mili Joule
NLFSR : Nonlinear-Feedback Shift Register
PDA : Personal Digital Assistant
PDF : Portable Document Format
PSNR : Peak Signal-To-Noise Ratio
RF :Radyo Frekansı
QoE : Quality of Experience
QoS : Quality of Service
RLC : Run-Length Coding
RLE : Run Length Encoding
RSA : Rivest–Shamir–Adleman
SPHIT : Set Partitioning in Hierarchical Trees
TIFF : Tagged Image File Format
VLC : Variable-Length Coding

1. GİRİŞ

Son yıllarda düşük maliyetli, düşük enerjili, çok fonksiyonlu, veri toplama ve işleme yeteneğine sahip küçük boyutlu akıllı algılayıcı düğümlerin geliştirilmesi dikkatleri bu konunun üzerine topladı [1]. Bu algılayıcılar bulunduğu alandan veri toplama, işleme ve belirli kararlar doğrultusunda elde ettiği bilgileri belirlenen bir merkeze gönderme kabiliyetine sahiptirler. Sahip oldukları bu özellikler sayesinde kablosuz algılayıcı düğümler birçok alanda kullanılmaktadır.

Algılayıcı düğümleri içerisinde kullanım amacına göre bir veya daha fazla algılayıcı, işlemci, bellek, güç kaynağı, radyo barındıran cihazlardır [2]. Ortamın özelliklerini ölçmek için çeşitli algılayıcılar bu düğümlere eklenebilir. Algılayıcı düğümler üzerinde sahip oldukları algılayıcılar ile ortamdan bilgi toplama, sahip olduğu donanımlar ile bu bilgiyi işlemeye, gerektiğinde bu bilgiyi RF (elektromanyetik radyo dalgaları) ortamında gönderebilme yeteneğine sahiptirler.

Kablosuz Algılayıcı Ağları (KAA), çok sayıda kablosuz algılayıcı düğümünden ve en az bir ana istasyondan (sink) oluşur [3]. Ana istasyon bir bilgisayar veya bir düğüm olabilir. Ana istasyon verinin işlendiği ve depolandığı merkezdir. Algılayıcı düğümler izlenilecek alana dağıtılır. Her bir düğüm veri toplayabilir ve elde ettiği veriyi ana istasyona gönderebilir. Algılayıcıların izlenilecek alanda, konumlarının önceden belirlenmesine gerek yoktur. Rastgele bir şekilde düğümler bu alana dağıtılabilir [3]. Şekil 1.1'de tipik bir KAA ve Kablosuz Multimedya Algılayıcı Ağı (KMAA) yapısı gösterilmiştir.



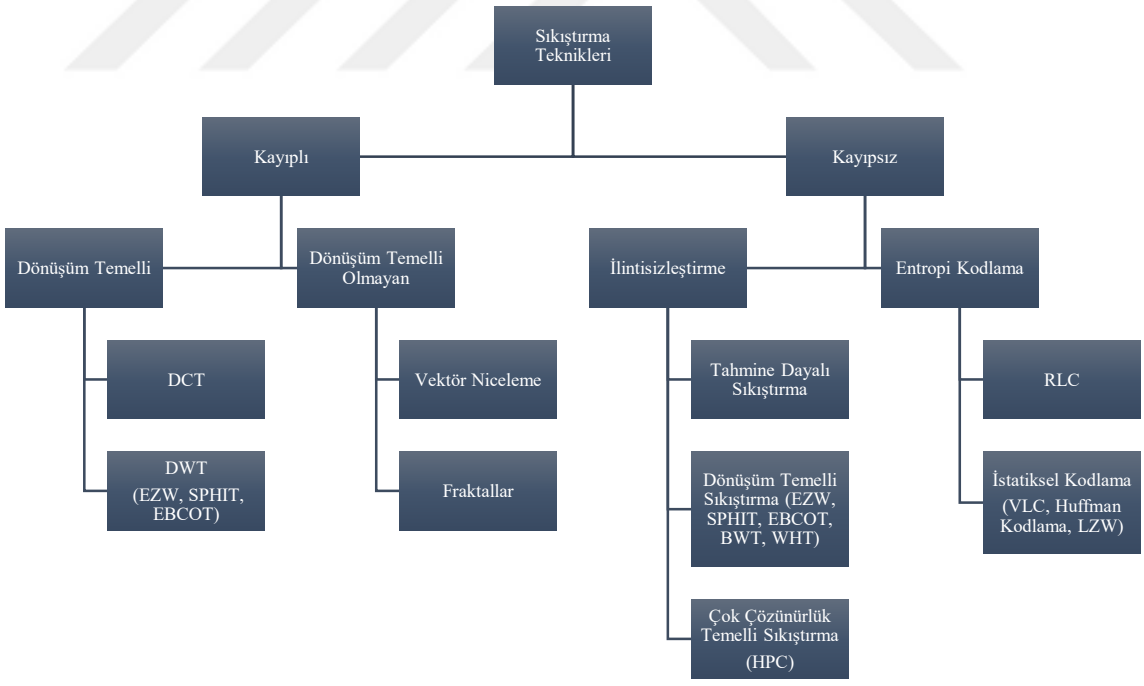
a) b)

Şekil 1.1. Tipik bir a) KAA yapısı [4] b) KMAAA yapısı [5]

Kablosuz multimedya algılayıcı ağları ise (KMAA), kablosuz algılayıcı ağlardan farklı olarak mikrofon, kamera gibi multimedya (çoklu ortam) içerik üreten algılayıcılar barındırırlar. KMAA'lar bulundukları ortamdan ses, video, resim ve sayısal veriler toplayabilme kabiliyetine sahiptirler [6]. KMAA'lar bulunduğu ortamdan aldıkları ses ve görüntü ile izlenilen ortam hakkında detaylara ulaşılmasını sağlar.

KAA izlenilen ortamdan aldığı düşük boyutlu sayısal verileri ana merkeze gönderirken; KMAA izlenilen ortamdan ses, video ve görüntü gibi yüksek boyutlu verileri ana merkeze göndermektedir. Gönderilen verinin büyük olması sınırlı enerji kaynağına sahip KMAA'ların yaşam ömürleri için büyük bir problem teşkil etmektedir. KMAA'larda algılanan verinin işlenmesi ve iletilmesi sürecinde harcanan enerji miktarını düşürmek, sistemin çalışma ömrünü arttırmak için kullanılan etkili yöntemlerin temelinde multimedya verisinin sıkıştırılması, yani veri boyutunun küçültülmesi yatmaktadır.

Büyük boyutlara sahip multimedya verilerinin saklanması ve iletilmesinde hafıza gereksinimlerini azaltmak, bant genişliğini etkin kullanmak ve enerji verimliliği için farklı sıkıştırma yöntemleri geliştirilmiştir. Görüntü sıkıştırma işlemi için birçok yöntem geliştirilmiş olup; klasikleşmiş bu yöntemlerin genel sınıflandırılması Şekil 1.2' de gösterilmiştir.



Şekil 1.2. Görüntü sıkıştırma yöntemleri [7]

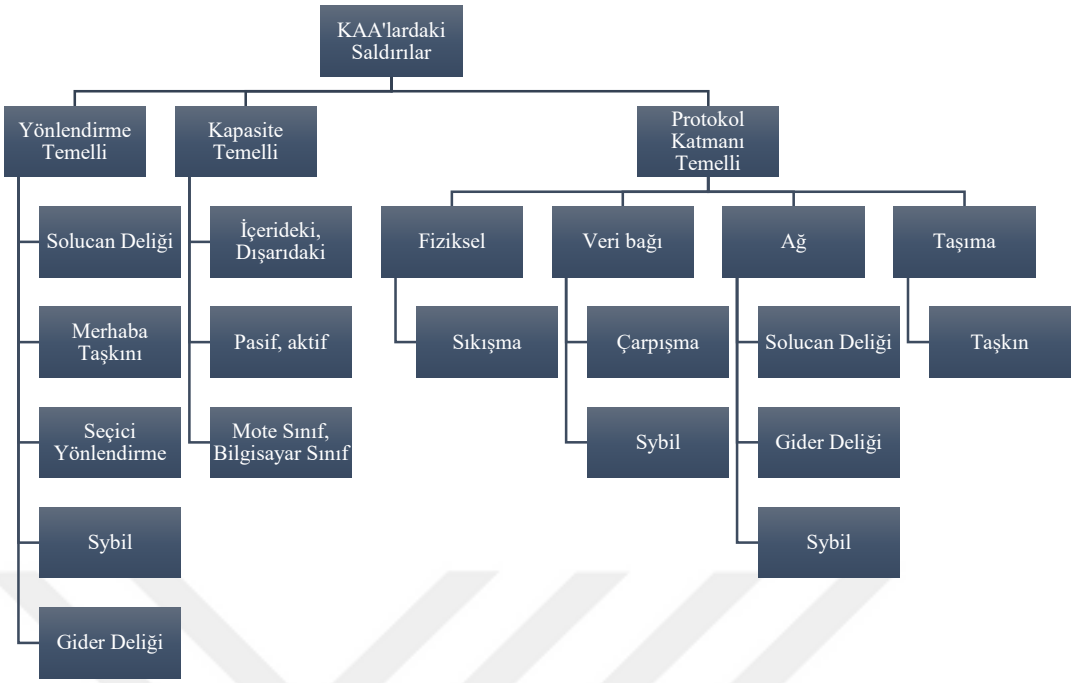
Şekil 1.2'den görüleceği üzere görüntünün sıkıştırılması için geliştirilen teknikler; kayıplı ve kayıpsız sıkıştırma olmak üzere iki sınıfa ayrılır. Kayıplı sıkıştırma yöntemlerinin kayıpsız

sıkıştırma yöntemlerine göre faydası: kodlama, kod çözme süresini azaltması, sıkıştırma oranı ve enerjidir [8].

KMAA'ların işletme doğasından ötürü uzun süreler boyunca çalışmaları için; gerek multimedya verisini ön işleme ve gerekse iletme süreçlerinde mümkün olduğu kadar az enerji harcamaları gerekir. Bundan dolayı Şekil 1.2'de gösterilen klasik resim sıkıştırma tekniklerinin, sıkıştırma sürecindeki matematiksel işlem yoğunluklarının fazla olması, yani fazla işlemci zamanı harcaması açısından KMAA'larda kullanılması fazla uygun olmamaktadır. Dolayısıyla KMAA'lar için daha az işlemci zamanı harcayan, daha basit işlemleri kullanan yeni görüntü sıkıştırma teknikleri üzerinde çalışmalar güncelliğini koruyarak devam etmektedir.

Bu konuda yapılan çalışmalardan bazıları; Clark N.Taylor [9] KMAA'ın çalışma ömrünü uzatmak için JPEG resim sıkıştırma algoritmasının resim niceleme ve ayrık kosinüs dönüşümü safhalarındaki parametreleri değiştiren bir metadoloji önermiştir. Abdelhamid Mammeri vd. [10,11], ayrık kosinüs dönüşümü (AKD) sürecinde resim sıkıştırma oranını arttıran ve enerji kullanımını azaltan uygulamalar önermiştir. Mammeri vd. [12] resmin tamamının sıkıştırılması yerine; ana istasyonda bulunan kullanıcının belirlediği alanlarda (verilen koordinatlarda) resmin kalitesini koruyabilmek için minimum sıkıştırma, diğer alanlarda ise maksimumum sıkıştırmaya dayanan bir yöntem önerdiler. Enyan Sun vd. [13] KMAA'nın ağ ömrünü uzatmak, AKD'de harcanan enerji miktarını düşürmek ve iletilecek veri miktarını azaltmak için bir algoritma önermiştir. Algoritma algılayıcı kamera tarafından alınan resmin, daha önce çekilen ve arka plan resmi olarak kullanılan resmin referans alınması suretiyle bir sıkıştırmaya tabi olmasına dayanır. Arafat Şentürk [14] KMAA'da sık kullanılan AKD, SPHIT ve LEICA algoritmalarının birbirlerine göre üstünlüklerini araştırmış ve gerçek zamanlı uygulamalar için LEICA algoritmasının uygun olduğu sonucuna varmıştır.

Öte yandan birçok KMAA ağları, askeri, tıbbi görsel bakım, endüstriyel işlem kontrolü, video gözetim sistemleri gibi güvenlik ve gizlilik gerektiren uygulamalar için de sıkça tercih edilmektedir. Ancak KMAA'lar genellikle uzak alanlarda kullanılırlar ve dışarıdan gelen tehditlere karşı savunmasızdırlar. KAA'lar için gerçekleştirilen saldırı tiplerinin ana sınıflandırması Şekil 1.3'te gösterildiği gibidir.



Şekil 1.3. KAA'lardaki saldırı türlerinin sınıflandırılması [15]

Kablosuz kanalda yayınlanan önemli verilerin istenmeyen kişiler tarafından ele geçirilmesini engellemek, verinin yetkisiz kişilerce değiştirilmeden hedef noktaya iletilmesini sağlamak, kimlik doğrulaması için ağ ve veri güvenliğinin sağlanması önemli bir problemdir. Üzerinde çalışılan önemli konulardan biridir. Sınırlı kaynaklara sahip KMAA'lar için yapılan güvenlik çözümleri, geleneksel güvenlik çözümlerine göre ek zorluklar oluşturur. KMAA'larda güvenliğin sağlanması için birçok güvenlik protokolleri geliştirilmiştir ve geliştirilmeye devam etmektedir. Bu algoritmaların bazıları, SPINs, LEAP, TINYSEC vb. dir [15].

Verinin yetkisiz kişilerce ele geçirilmesinin engellemek amacıyla yapılan bazı çalışmalar şöyledir: Lim Chong HAN ve Nor Muzlifah MAHYUDDİN [16] özel veya (XOR) şifreleme ve Sezar şifrelemenin bir kombine halini kablosuz haberleşme için geliştirdiler. Bu uygulamanın işlem zamanını etkilemediği ve güvenli bir kablosuz haberleşme sağladığını kanıtladılar. Rad R. MORADİ vd. [17] resmi belirli boyuttaki (8 X 8 piksel grubu) bloklara ayırıp; bu blokları tanımlı çeşitli modellere göre karıştırdılar. Karıştırılan ve algılanamayan görüntü, bölümlere ayrılarak (örneğin dört bölüme ayrılarak); bu bölümlerin karşılık gelen blokları arasında XOR işlemi gerçekleştirilmesi esasına dayanan bir yöntem önermişlerdir. V.V. DİVYA vd. [18] AKD uygulanmış bir resim verisinde seçilen katsayılar üzerine bir rastgele bit üreticisinden gelen bit ile XOR işlemi gerçekleştirilmesine dayanan bir yöntem önerdiler. Lala KRIKOR vd. [19] AKD uygulanan bir resim verisi üzerinde yüksek frekanslı bazı katsayıları bit üreticisinden gelen (NLFSR + anahtar) bit akışı ile XOR işlemine tabi tutarak resmi şifrelediler. Shu-Jiang Xu vd. [20]

kaos temelli dairesel bit kaydırma ve XOR işlemlerini kombine eden bir kaotik şifreleme sistemi önerdiler.

KMAA'larda veri şifrelemesinin getireceği faydanın yanı sıra genellikle zaman gecikmesi, başarılı teslimat ve güç tüketimi gibi diğer ağ performansları için bir maliyeti olduğu da göz ardı edilemez. Bu nedenle, sağlanan gizlilik düzeyi ve diğer performans ölçümleri arasında, özellikle de baştan sona gecikme ve enerji tüketimi arasında bir dengenin oluşması son derece önemlidir [21].

1.1. Tezin Amacı

KMAA'lar askeri alanlardan, sağlık alanına, habitata kadar çok farklı sektörlerde büyük uygulama alanı bulmuştur. Sahada kullanılan ve ağ ömürlerinin yüksek olması istenen KMAA düğümlerinin en önemli dezavantajları; besleme enerjilerinin, işlemci ve hafıza kapasitelerinin limitli olmasıdır. Dolayısıyla, büyük boyutlu çoklu ortam verilerinin klasik yöntemlerle işlenmesi sürecindeki işlem yükü, hafızaya olan ihtiyacı, iletişim için yüksek bant genişliğine gerek duyulması, KMAA'larda bu verilerin oluşturulması ve iletilmesi sürecindeki önemli iyileştirmelerin yapılmasını gerektirir. Ayrıca gizlilik ve özellik arz eden multimedya verilerinin kablosuz ağ üzerinden iletilmesi sürecinde yetkisiz kişiler tarafından verinin ele geçirilmemesi, değiştirilmemesi veya okunmaması da önemli problemlerden bir tanesidir. Dolayısıyla KMAA'ların kısıtları nedeniyle güvenli veri iletişimi için, klasik şifreleme algoritmalarının dışında yeni güvenlik protokol ve algoritmalarının geliştirilme gerekliliği de bir diğer güncel problemdir.

Görüntü sıkıştırma ve şifreleme yöntemlerindeki her türlü iyileştirme ve geliştirmenin KMAA'ların pratik hayatta hemen her sektörde kullanım alanını arttırdığı bir gerçektir.

Bu tezin amacı; KMAA'lar da, resim oluşturma ve iletme süreci için çalışılmış sıkıştırma ve şifreleme yöntemlerinin ve iletişim protokollerinin incelenerek, KMAA'lar için daha az enerji harcayan, işlem kapasitesi düşük mevcut kayıplı veri sıkıştırma ve şifreleme tekniklerinde iyileştirme sağlayabilecek öneriler ortaya koyabilmek ve bunun gerçekleştirilebilir olabileceğini göstermektir.

Bu tez çalışmasında;

- Gri seviye resim kareleri üzerinde klasik JPEG kayıplı sıkıştırma algoritmasının detaylı bir şekilde incelenip; kullanılacak kayıplı resim sıkıştırma altyapısının hazırlanması,
- Literatür taramasının yapılarak KMAA'larda görüntü sıkıştırma için önerilen mevcut algoritmaların denenmesi ve eksikliklerinin tespit edilip yapılabilecek iyileştirmeler için ek algoritmalar önerilmesi, test edilmesi, mevcut algoritmaya göre üstünlüklerinin gösterilmesi, hesaplama zamanının elde edilmesi,

- KMAA'larda güvenli veri aktarımı için uygun ve basit bir şifreleme modelinin oluşturulması ve hesap zamanı açısından uygunluğunun gösterilmesi,

için MATLAB ortamında geliştirilen uygulama programları, sonuçları ve değerlendirmeler açıklanarak, önerilen resim sıkıştırma ve şifreleme sisteminin orta sınıf veya PDA sınıf platform seviyesindeki KMAA düğümlerinde kullanılabilir olması gözetilmiştir.

1.2. Tezin Organizasyonu

Bölüm 2'de KMAA'lar, Bölüm 3'te görüntü sıkıştırma, Bölüm 4'te KMAA için JPEG uyarlamalarının incelenmesi, Bölüm 5'te veri şifreleme hakkında bilgiler verilmiştir. Bölüm 6'da önerilen resim sıkıştırma ve şifreleme yöntemi sunulmuştur. Bölüm 7'de bulgular verilmiş olup son bölümde sonuçlar ve öneriler verilmiştir.

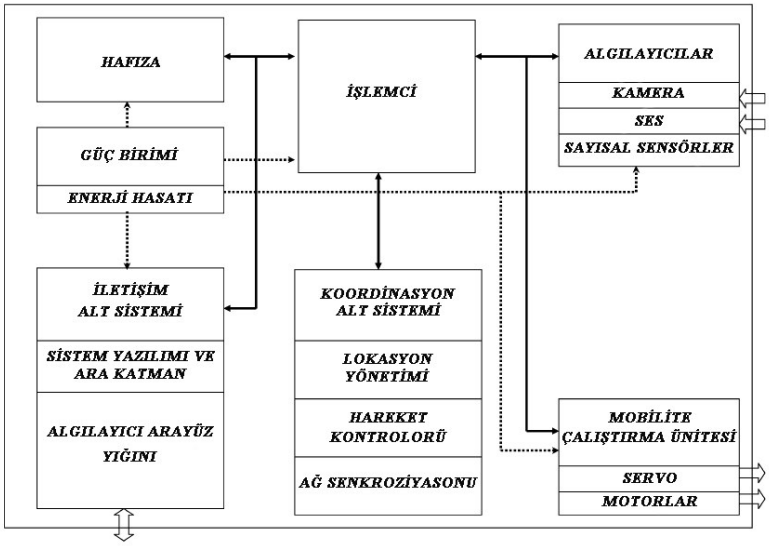
2. KABLOSUZ MULTİMEDYA ALGILAYICI AĞLAR

Bir KAA; ilgilenilen hedef bölgeden toplanan veriyi, kablosuz bir şekilde iletebilen algılayıcılardan oluşur. Belirli bir alanda çoğunlukla gözlem, takip ve izleme yapmak için kullanılan KAA'lar, uygulamaya bağlı olarak gerekli algılama birimlerini (sayısal sensörler), işlemci, hafıza, güç ünitesi ve kablosuz iletim birimini içermelidir. KAA'lardaki düğümlerin belirlenmiş zamanlarda veya bir olayın tetiklemesi anında uyanması, algılayıcılarını ölçmesi ve verisini gönderdikten sonra tekrar uyuma pozisyonuna geçmesi şeklindeki çalışması bu ağlardaki enerji sarfiyatının azaltılması için gerçekleştirilen önemli bir çalışma şeklidir. Öte yandan KAA'lardaki iletilecek sayısal sensör verisi düşük kapasiteli olduğundan iletim sürecindeki enerji harcaması da çok fazla olmayacaktır.

Buna karşılık KMAA'lar çalışma prensibi olarak KAA'lara benzerlik göstermekle birlikte, çoklu ortam algılayıcılarından elde edilen büyük kapasiteli verinin ön işlenmesi ve iletilmesi süreçlerinde önemli miktarda enerjiye ihtiyaç duyabilmektedir. Enerji kullanımını azaltabilmek için farklı yapılı KMAA düğümleri, topolojileri, iletişim ve güvenlik protokolleri geliştirilme çalışmaları devam etmektedir. Bu bölümde KMAA'lar hakkında genel yapısal açıklamalar verilecektir.

2.1. KMAA Algılayıcısının Yapısı

KMMA'lar üzerindeki sayısal sensörlerden (algılayıcılar) ayrı olarak dahili olarak barındırdıkları kamera, mikrofon ile izleme yaptıkları çevreden ses görüntü vs. gibi multimedya verilerini toplayabilirler. Tipik bir KMAA'nın yapısı Şekil 2.1'de verilmiştir.

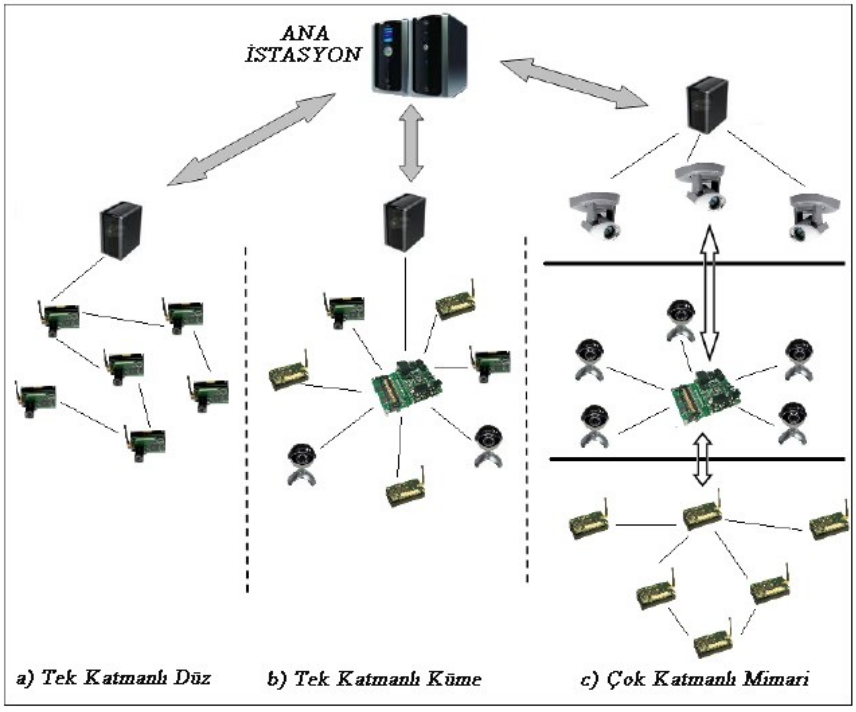


Şekil 2.1. Tipik bir KMAA'nın iç yapısı [5]

Bir KMAA Şekil 2.1'de gösterildiği üzere çeşitli birimlerden oluşur [5]. Bunlar algılama birimi, işlemci, iletişim sistemi, koordinasyon sistemi, hafıza ve tercihe bağlı olarak hareket/çalıştırma birimidir. Algılama birimleri iki birimden oluşur: algılayıcılar (kamera, mikrofon veya sayısal algılayıcılar) ve analog dijital dönüştürücüler. ADC, analog bir sinyali sayısal sinyale çeviren birimdir. İşlemci, algılama ve iletişim görevlerinin sorumluluğunda sistem yazılımlarını çalıştırır. İletişim sistemi bir telsiz ve iletişim yazılımından oluşur. Bir koordinasyon sistemi; ağ senkronizasyonu ve lokasyon yönetimi gibi işlemler gerçekleştirerek farklı ağ cihazlarının farklı işlemlerini düzenlemeden sorumludur. Tercihe bağlı olan hareket/çalıştırma birimi, nesnelerin hareketine olanak tanır. Sisteme bir enerji ünitesiyle enerjisi verilir.

2.2. KMAA'larda Ağ Mimarileri

KAA'larda önerilen ağ mimarilerin birçoğu heterojen olarak dağıtılmış algılayıcıların oluşturduğu düz bir mimaridir [21]. KMAA'ların ortaya çıkışıyla birlikte, ağın belirli uygulama ve hizmet kalitesi(QoS) gereksinimlerine bağlı olarak daha ölçeklenebilir ve daha verimli olabileceği şekilde ağı farklı mimarilere yeniden yapılandırma ihtiyacı vardır. Bu tasarlanan ağ topolojisi temelinde; multimedya içeriklerinin işlenmesi ve iletimi için yönlendirme sürecinde ağdaki mevcut kaynaklardan eşit ve verimli bir şekilde yararlanılması, kullanılması yatmaktadır. Şekil 2.2'de KMAA'larda kullanılan ağ mimarileri gösterilmiştir.



Şekil 2.2. KMAA'larda ağ mimari modelleri [21]

Şekil 2.2'den de görüleceği üzere ağ mimarileri üç farklı modele ayrılmıştır [21]. Tek katmanlı düz mimari: aynı kabiliyetlere ve fonksiyonlara sahip algılayıcılarla uygulanmıştır. Bu modelde tüm düğümler; multimedya işleme yoluyla görüntü yakalamadan, çoklu atlama temelinde ana istasyona doğru veri aktarımına kadar herhangi bir işlevi gerçekleştirebilir. Bu mimarinin yönetimi kolaydır. Ayrıca multimedya işleme, algılayıcılar arasında dağıtılmıştır ki bu ağ ömrünü uzatır. İkinci model olan tek katmanlı küme ise her küme içindeki kamera, ses ve sayısal sensörlerin verileri bir küme başına aktardığı heterojen sensörlerle uygulanmıştır. Küme başının daha fazla kaynakları vardır ve daha yoğun veri işleme kabiliyetine sahiptir. Küme başı toplama merkezine veya bir çıkış kapısına direkt veya başka küme başları vasıtasıyla çok atlamalı tarzda kablosuz bir şekilde bağlıdır. Üçüncü model olan çok katmanlı mimaride ise, hareket tespiti gibi basit görevleri gerçekleştirmek için ilk katman sayısal algılayıcılar ile, kamera algılayıcılarının ikinci katmanı nesne tespiti, nesne tanıması gibi daha karmaşık görevlerine yapabilir, üçüncü katman daha güçlüdür ve yüksek çözünürlük kamera algılayıcıları, nesne izleme gibi daha fazla karmaşık işleri yapabilir.

2.3. Kablosuz Algılayıcı Düğümlerin Sınıflandırılması

Kablosuz algılayıcılar işleme güçleri ve depolama kapasitelerine göre üç sınıfa ayrılabilir [21]:

- **Düşük Sınıf Platformlar:** Isı, ışık nem vb. gibi sayısal verileri tespit etmek için tasarlanan bu kategorideki cihazların ana amacı mümkün olduğunca az enerji tüketmektir. Düşük bir işleme kabiliyetine, küçük bir hafızaya ve çoğunlukla temel bir iletişim yongasına (örneğin, CC2420 radyoda IEEE 802.15.4) sahiptirler.
- **Orta Sınıf Platformlar:** Bu sınıfta bulunan cihazlar düşük sınıf platformlara göre daha iyi bir işleme gücüne ve depolama kapasitesine sahiptirler. Ancak bunlar düşük bant genişliği ve veri hızı iletişim modülü ile donatılmışlardır. Düşük sınıflı platformlara göre daha fazla enerji tüketirler.
- **PDA Sınıf Platformlar:** Bu kategorideki cihazlar hesaplama ve işlem gücü açısından daha güçlüdürler ve hızlı ve verimli bir şekilde multimedya içeriklerini işlemek için tasarlanmışlardır. Bu cihazlar farklı işletim sistemlerini çalıştırabilirler ve farklı veri hızları ile çoklu radyoları desteklerler.

Şekil 2.3'te kablosuz algılayıcı platformlar için bazı örnekler gösterilmiştir.

Şekil 2.3. Kablosuz algılayıcı platform örnekleri [21]

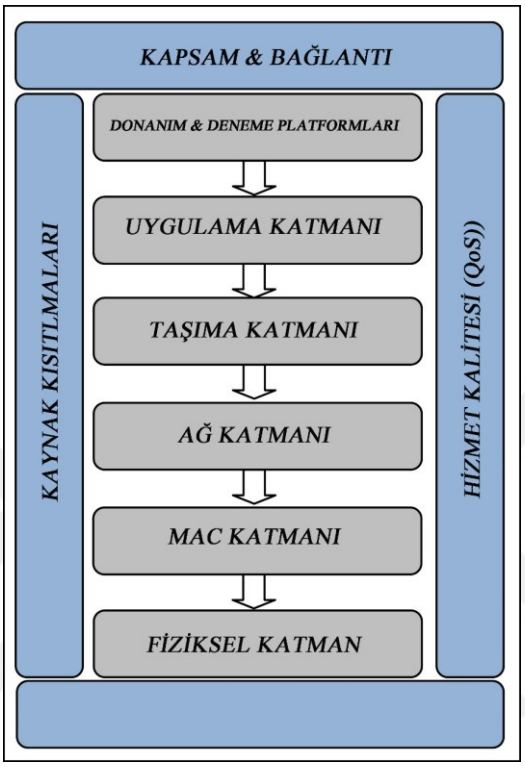
2.4. Çözünürlüklerine Göre Kamera Sınıfları

Kablosuz algılayıcılarda kullanılan kameraların türleri, uygulamanın önemine göre farklılık göstermektedir [22]. Bu kameraların farklı kabiliyetleri (çözünürlük, işleme gücü, depolama, vd.) vardır. Kabiliyetlerine göre ağda farklı görevler üstlenirler. Çok katmanlı bir ağın düşük katmanında; düşük çözünürlüklü kameralar, düşük enerji harcama özelliklerinden yararlanılarak basit nesne tespiti için kullanılabilirler. Cyclops, CMUCam3, ve eCam düşük çözünürlüklü kameralara örnektir. Orta ve yüksek çözünürlüklü kameralar, nesne tanıma ve izleme gibi daha karmaşık ve güç tüketen görevler için ağın daha yüksek katmanlarında kullanılabilir. Bu tür kameralar daha fazla güç tüketir ve bu nedenle bir nesneyi tespit etmek durumunda, yalnızca daha düşük katmanlı cihazlar tarafından talep üzerine uyanır. Şekil 2.4'te çözünürlüklerine göre kamera örnekleri gösterilmiştir.

Şekil 2.4. Çözünürlüklerine göre kamera türleri [22]

2.5. KMAA'da Katmanlı Yapı

KMAA'larda beş katmanlı bir yapı bulunmaktadır. Bu durum Şekil 2.5'te gösterildiği gibidir.



Şekil 2.5. KMAA'lardaki katmanlı yapı [22]

Fiziksel katman, bir ağın temel donanım iletim teknolojilerini barındırır ve ağ düğümlerini bağlayan kablosuz bağlantı üzerinden mantıksal veri paketleri yerine ham bitleri iletme yollarını tanımlar [22]. Ayrıca frekans seçimi, modülasyonu ve kanal kodlamasından da sorumludur.

Yüksek verimli ve güvenilir orta erişim kontrol (MAC) protokollerinin tasarımı, kablosuz sensör ağında kritiktir [22]. Geleneksel olarak amaç, ılımlı bir ağ yükü koşulu altında minimum enerji maliyetinde yeterli iletim kapasitesi sağlamaktır. Bu nedenle, KMAA'lar için MAC katman protokolü aşağıdaki özellikleri karşılamalıdır:

- Ağ verimliliğini en üst düzeye çıkarmak,
- İletim güvenilirliğini artırmak,
- Kontrol yükünü en aza indirgemek,
- Enerji tasarruflu olmak,
- Belli bir hizmet kalitesi(QoS) seviyesini garanti eder.

KAA'daki ağ katmanı, enerji verimliliği, bağlantı kalitesi, hata toleransı ve ölçeklenebilirlik gibi çeşitli tasarım hususlarını dikkate alarak algılanan verileri kaynaklardan toplama merkezine

iletmeyi amaçlamaktadır [22]. Geleneksel KAA için önerilen birçok yönlendirme protokolü olmasına rağmen, KMAA için yönlendirme protokollerinin tasarımı hala aktif bir araştırma alanıdır. Ağ üzerinden multimedya içeriği işlemesi nedeniyle KAA'lar için önerilen yönlendirme protokollerini KMAA'lar için doğrudan uygulanamayacağı kanaatine varılmıştır.

Taşıma katmanı, uçtan uca mesaj iletimini etkinleştirmek için ağ katmanı üzerinde çalışan bir protokoldür [22]. Taşıma katmanı; aynı sipariş teslimatı, veri güvenilirliği ve kaybının geri kazanımı, akış ve tıkanıklık kontrolü ve muhtemelen hizmet kalitesi (QoS) gereklilikleri gibi çeşitli hizmetleri sağlamayı amaçlamaktadır.

KMAA'daki uygulama katmanı heterojen işlevler sağlar ve aşağıdakileri hizmetleri destekler [22]:

1) Uygulamaya özgü gereksinimlere ve donanımın kapasitesine bağlı olan multimedya işleme ve kaynak kodlama teknikleri.
2) Ortak ağ içi multimedya işleme mekanizmalarını desteklemek için ağ üzerindeki diğer uygulama programları ile etkili iletişim.
3) Trafik yönetimi ve kabul kontrolü.

2.6. KMAA'larda Veri Yönlendirme Prensipleri

Hem KAA'lar hem de KMAA'lar genellikle pille çalışan düğümler kullanırlar ve bu nedenle yönlendirme protokollerinin tasarımında enerji optimizasyonu konusunda ortak bir dezavantajları vardır. Bu benzerliklerin yanı sıra, özellikle multimedya destek yetenekleri açısından da temel farklılıklar vardır. İlk olarak, tipik KAA yönlendirmesi, yüksek bant genişliği, yüksek doğruluk ve işlem ve gerçek zamanlı iletim enerjisi gerektiren multimedya uygulamaları için tasarlanmamıştır. KMAA'larda, heterojen sensörler devreye alınabileceğinden, yönlendirme için ek zorluklar getiren video akışları, hareketsiz görüntü, ses ve sayısal veriler kullanılabilir. İkincisi, bir yönlendirme kararı verirken enerji verimliliği ile multimedya QoS / QoE arasında bir denge vardır. Veri toplama ve sıkıştırma için yönlendirme şemaları enerji tasarrufu için yaygın olarak uygulanır. Ancak KMAA'larda kabul edilemez bir gecikmeye neden olabilirler. Üçüncüsü, multimedya algılama ve dağıtım modeli, özellikle enerji tüketim oranı ve rota değişikliklerinin sıklığı için yönlendirme performansını etkiler, yani bir düğüm sürekli olarak bir multimedya içeriğini yakalayıp iletirken; daha fazla enerji tüketilir ve bu nedenle yollar sürekli olarak hesaplanmalıdır. Yaygın olarak kullanılan MPEG-4 önemli enerji tüketimine yol açar ve bu nedenle yönlendirme verimliliğinde bozulmaya neden olur. Bu kısıtlamalar ve zorluklar, multimedya uygulamalarının bant genişliğine ve gecikmeye duyarlı doğası ile birlikte; KMAA'larda yönlendirme işlemini oldukça zor ve çözülmesi gerekli bir problem şekline getirir. KMAA'larda mevcut yönlendirme çözümleri beş ana kategoride sınıflandırılmaktadır: Bunlar sırasıyla;

1. QoS sağlama,

2. Multimedya farkındalığı (tip seçiciliği),
3. Enerji verimliliği,
4. Tıkanıklıktan kaçınma,
5. Bant genişliği optimizasyonu.

Her kategoriye ait yönlendirme çözümleri, tasarım ve optimizasyon hedeflerine göre ayrıca sınıflandırılır. Beş kategorinin her birine ait örnek yönlendirme teknikleri sırasıyla, gecikme garantili yönlendirme, diferansiyel kodlama tabanlı multimedya yönlendirme, görev döngüsü tabanlı gecikmeye duyarlı yönlendirme, potansiyel alan tabanlı yönlendirme, spektrum etkin multimedya yönlendirme olarak verilebilir [23].

3. GÖRÜNTÜ SIKIŞTIRMA

Resim: bir varlığın görünüşünün bir madde (kağıt, bez vb.) üzerine çeşitli araçlar ile (kalem, fırça vs.) ile çizilmesidir. Bilinen en eski resmin tarihi yaklaşık 65.000 yıl öncesine dayanmaktadır [24]. İnsanlık tarihi boyunca insanlar, önemli gördükleri olayları resmetmiştir. Geçmişte yaşayan insanların yaşam biçimleri, tarihsel olayların gerçekleşme biçimi hakkında bilgiler yine o tarihlerde yapılmış resimlerden yararlanarak elde edilmektedir.

Kelime anlamı Yunanca'dan 'Photos' 'ışık' ve 'Graphos' 'çizmek' kelimelerinin bitişik yazılmasından oluşan fotoğraf; optik ve kimyasal süreçlerin kullanılmasıyla yüzey üzerine görüntü elde etmek olarak tanımlanabilir [25]. Bilinen ilk fotoğraf 1826 yılında çekilmiştir. Fotoğraf sadece 30 yıl içerisinde herkesin kullanabileceği bir ürün haline geldi [26]. Fotoğrafın ortaya çıktığı ilk günden bu güne kadar geçen süre içerisinde teknoloji hızla gelişme kaydetmiştir. Gelişen teknolojiyle birlikte yüksek çözünürlüklü, yüksek boyutlara sahip fotoğraflar çekebilen cihazlar üretilmiştir. Yüksek boyutlu fotoğrafların çekilmesi, sınırlı kaynaklara sahip bilişim cihazlarında saklanması problemini doğurmuştur. Bu sınırlı kaynaklarda daha fazla fotoğraf nasıl saklanır sorusunun çözümü için çeşitli yöntemler geliştirilmiştir. Bu yöntemler resim sıkıştırma yöntemleri olarak adlandırılmaktadır.

3.1. Veri Sıkıştırma Yöntemleri Tarihçesi

Veri sıkıştırma yöntemlerinden olan ve bilinen ilk örneklerinden olan mors kodlaması İngilizcede sıkça kullanılan 'e' ve 't' gibi harflerin daha kısa bir şekilde kodlanmasına dayanan yöntemdir [27].

Shannon-Fano algoritması [28] 1949 yılında geliştirilmiş olup; bilinen sembollerin olasılıklarını hesaplamakta ve onları iki kümeye bölmektedir. İki kümedeki sembollerin olasılıklarının birbirine yakın olması bu algoritmada sağlanır. İlk kümedeki sembollerin başlangıcı sıfır, ikinci kümedeki sembollerin başlangıcı bir değeri verilir. Bu bölünmüş kümeler içinde olasılıkları birbirine yakın iki küme oluşturulur ve ilk kümeye sıfır ikinci kümeye bir değer verilir. Bu işlem her bir sembole karşılık bir kod elde edilinceye kadar devam eder ve bir ağaç yapısı oluşturulur.

David A. Huffman tarafından 1952 yılında geliştirilen Huffman kodlaması [28] Shannon-Fano algoritmasından farklı olarak sembollerin olasılıklarına göre kökten yaprağa bir ağaç oluşturmak yerine; yapraktan köke doğru bir ağaç oluşturmayı amaçlayan algoritmadır. Shannon-Fano algoritmasına göre avantajı her zaman en uygun ağacı oluşturmasıdır.

Lempel ve J. Ziv tarafından LZ77, LZ78 algoritmaları sırasıyla 1977 ve 1978 yıllarında geliştirilmiştir [29]. Bu algoritmalar sözlük temellidir. LZ77, bir kayan pencere üzerinden daha

önce geçen karakterlere göre kodlama veya kod çözme işlemini gerçekleştirir. LZ77 şifreleme süreci, eşleşen modeli bulmak için çok sayıda karşılaştırma gerektirdiğinden zaman alıcıdır [30]. LZ77 algoritmasının harici bir sözlüğü olmadığı için; algoritma herhangi bir eşleşme bulamadığında, daha fazla hafıza kullanan ve algoritma uygulanma süresini arttıran uzunluk ve ofset olarak kodlar[30]. LZ77önceki verilere göre çalışırken; LZ78 sonraki verilere göre çalışır. LZ78 İleri tarama ile tuttuğu bir sözlüğe göre bir arabellekte bulunan verileri eşleştirerek çalışır. Sözlükte herhangi bir eşleşme kalmayıncaya kadar tüm arabelleği tarar.

Terry Welch tarafından 1984 yılında LZ78 algoritmasının geliştirilmesiyle LZW algoritması oluşturulmuştur [31]. Bu algoritma sözlük tabanlı olup bir görüntü ya da metin içerisinde tekrar eden veri gruplarını sözlükte bulunan bir kod ile değiştirir. Bu algoritma gelen veri eğer sözlüğünde yoksa bu veriyi sözlüğe ekler ve eklenen veri için bir kod oluşturur. Unix işletim sisteminde dosya sıkıştırma da, GIF resim formatında ve isteğe bağlı olarak PDF ve TIFF formatlarında da kullanılır [32].

1988 yılında diğer video kodlama standartlarının temelini de oluşturan ilk video kodlama formatı, 1989 ses verisinin sıkıştırılması için MP3 formatı oluşturuldu [33]. 1989 yılında birden fazla dosya veya klasörü barındırabilen ZIP dosyalarını oluşturan ve birkaç sıkıştırma algoritmasının kullanılmasına olanak sağlayan ZIP dosya formatı oluşturuldu [34]. 1992 yılında JPEG standardı oluşturuldu.

3.2. JPEG

Günümüzde gerek mobil telefonlardan, gerekse endüstriyel kameralardan alınan fotoğraflar sıklıkla JPEG formatında olup; üretilen görüntü alma cihazlarının çoğunun varsayılan ayarları da JPEG formatındadır [35]. Ayrıca internet ortamında kullanılan resimlerin çok büyük bir çoğunluğunun dosya biçimi de JPEG'dir. Farklı oranlarda sıkıştırma sağlayabildiği ve bu sebeple internet bant genişliğinin düşük olduğu ortamlarda bile hızlı bir şekilde resim aktarılmasını mümkün kıldığı için, JPEG hala çok popülerdir.

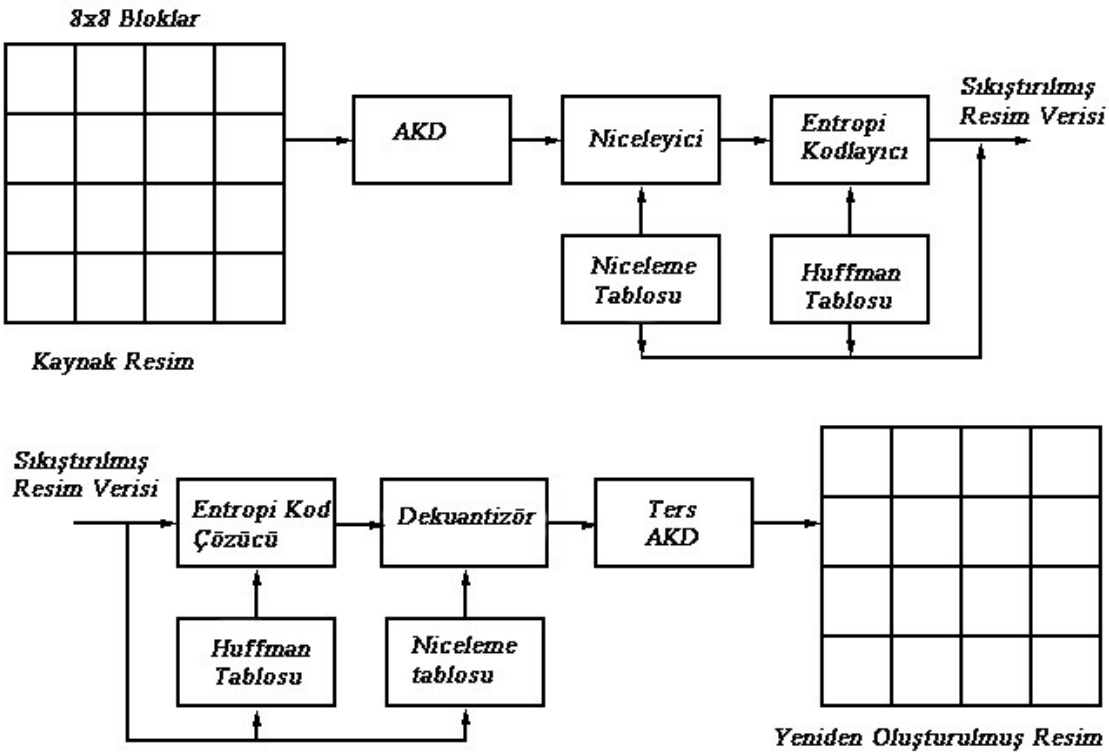
Joint Photographic Experts Group (Birleşmiş Fotoğraf Uzmanları Grubu) kelimelerinin ilk harflerinin birleşmesinden oluşan JPEG 1992 yılında ortaya çıktı. JPEG orijinal resimde bulunan bütün veriyi barındırmadığı için kayıplı bir resim sıkıştırma formatıdır. Resmin sıkıştırma oranı arttıkça buna bağlı olarak resmin boyutu azalmakta, fakat resmin kalitesi de düşmektedir. Eğer bir resim çok fazla sıkıştırılırsa gözle görülür şekilde resimdeki veri kayıpları fark edilebilir. Sıkıştırma oranı resmin görüntü kalitesi ve boyutu göz önüne alınarak değiştirilebilir. JPEG 10-12 aralığında sıkıştırma oranı sağlayarak; resmin saklanması için gerekli hafızadan kazanç sağlamamızı sağlar [35]. Resmin yüksek ve düşük kaliteli kopyalarını tutabilmesi ona büyük bir avantaj sağlamaktadır [36].

Şekil 3.1'de 512 X 512 boyutunda 24 bitlik sıkıştırmaya tabi olmayan, resim işleme uygulamalarında kullanılan standart orijinal Lena fotoğrafı gösterilmiştir. Şekil 3.2'de ise yine aynı boyut ve bit değerlerine sahip aynı resmin %50 sıkıştırma oranı, 4:2:0 alt örneklemeyle JPEG formatında gösterimi verilmiştir [35].

Şekil 3.1. Orjinal Lena 769 Kb [35].

Şekil 3.2. JPEG ile sıkıştırılmış Lena 23 Kb [35].

Şekil 3.2'deki sonuçtan da görüleceği üzere JPEG ile yaklaşık 33 kat sıkıştırma sağlanmasına rağmen iki resim arasında gözle görülür büyük bir farklılık yoktur [35]. JPEG çeşitli aşamalardan oluşur. Bu aşamalar Şekil 3.3'te gösterilmiştir.



Şekil 3.3. JPEG aşamaları [40]

3.2.1. Aralık Ortası

JPEG standardında bir resmin sıkıştırılması için ilk olarak resim bloklara bölünür. Blok 8 X 8'lik piksel grubundan oluşur. JPEG blok bazlı çalışan bir yöntemdir. Her bir blok verisi için JPEG'nin süreçleri sırasıyla işletilir.

Gri seviye bir resimde bulunan piksellerin değerleri 0-255 arasında değişmektedir. Bir kosinüs dalgası +1 ile -1 arasında değerler alır. İlk JPEG süreci olan ayrık kosinüs dönüşümün bir bloğa uygulanabilmesi için; bu blok verisine sıfır merkezli (kosinüs dalgasından dolayı) bir kaydırma işlemi gerçekleştirilir. Bu kaydırma işlemi blok içerisindeki piksel değerlerinden 128 değerinin çıkarılmasıyla gerçekleştirilir. Böylece elde edilen yeni veri -128,+127 aralığında değerlere sahip olur. Şekil 3.4'te rastgele alınan bir bloğa ait piksel değerleri verilmiştir.

	1	2	3	4	5	6	7	8
1	164	165	165	166	164	166	168	169
2	165	165	165	165	165	165	168	168
3	166	166	165	165	165	166	168	166
4	166	166	166	165	166	165	166	167
5	166	166	166	166	165	166	167	167
6	167	167	167	166	166	167	167	167
7	165	167	166	166	168	166	167	166
8	166	167	167	167	167	167	169	167

Şekil 3.4. Rastgele alınan bir bloğun piksel değerleri

Şekil 3.4'te verilen blok verisi için bütün JPEG süreçleri sonraki bölümlerde gerçekleştirilecektir. Şekil 3.5'te ise Şekil 3.4'te verilen veri bloğuna aralık ortası işleminin uygulanması sonucunda oluşan veri bloğu gösterilmiştir.

	1	2	3	4	5	6	7	8
1	36	37	37	38	36	38	40	41
2	37	37	37	37	37	37	40	40
3	38	38	37	37	37	38	40	38
4	38	38	38	37	38	37	38	39
5	38	38	38	38	37	38	39	39
6	39	39	39	38	38	39	39	39
7	37	39	38	38	40	38	39	38
8	38	39	39	39	39	39	41	39

Şekil 3.5. Aralık ortası işleminin uygulanması

3.2.2. Ayrık Kosinüs Dönüşümü

JPEG resim sıkıştırma süreci için anahtar niteliğinde Ayrık Kosinüs Dönüşümü (AKD), uzaysal alanda bulunan resim piksellerini frekans alanına dönüştüren bir tekniktir [37]. 2 Boyutlu AKD denklemi Denklem 3.1'de gösterilmiştir [38].

$$D(i,j) = \frac{1}{\sqrt{2N}}\, c(i)\, c(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y)\, \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \tag{3.1}$$

Denklem 3.1'de bulunan c(i) ve c(j) fonksiyonları Denklem 3.2'de gösterildiği gibi hesaplanır [38].

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}}, & u = 0 \\ 1, & u > 0 \end{cases} \tag{3.2}$$

Denklem 3.1'de p matrisi, resmin piksel değerlerinin matris halini göstermekte olup; p(x, y) resmin x, y indeksinde bulunan değeri ifade eder. AKD'de kullanılan blok boyutu N değeri olup; Denklem 3.1 p matrisinden, i, j'ninci dönüşüm değerini hesaplar. JPEG standart olarak 8 X 8 boyutundaki bloklar kullandığı için N değeri 8'e eşittir. Denklem buna göre tekrar düzenlenirse; bu durumda hesaplama Denklem 3.3'te gösterildiği gibi olur [38].

$$D(i,j) = \frac{1}{4}\, c(i)\, c(j) \sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y)\, \cos\left[\frac{(2x+1)i\pi}{16}\right] \cos\left[\frac{(2y+1)j\pi}{16}\right] \tag{3.3}$$

İki boyutlu AKD formülü Denklem 3.4'te gösterildiği gibi de ifade edilebilir [13].

$$D(k\ x\ k) = A(k\ x\ k)P(k\ x\ k)A^{T}(k\ x\ k) \tag{3.4}$$

Genellikle k değeri 8 olup; P(k x k) matrisi resmin piksel değerlerini içerir. A(k x k) AKD matrisini, A^T; A matrisinin transpozu olan matris ve AKD dönüşümünün sonucunda oluşan AKD katsayı D (k x k) matrisidir. Bir AKD matrisi 8 X 8 boyutunda olup 64 adet AKD katsayısı içerir.

AKD Matrisi

Denklem 3.1'den AKD matrisinin elde edilmesi için Denklem 3.5 kullanılır [38].

$$T(i,j) = \begin{cases} \frac{1}{\sqrt{N}}, & i = 0 \\ \sqrt{\frac{2}{N}} \cos\left[\frac{(2j+1)i\pi}{2N}\right], & i > 0 \end{cases} \tag{3.5}$$

Denklem 3.5'in 8 X 8 boyutundaki bir blok için hesaplanmasının Şekil 3.6'da gösterilmiştir. Şekil 3.6'da elde edilen sonuç, Denklem 3.4'te kullanılan A matrisidir.

	1	2	3	4	5	6	7	8
1	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536	0.3536
2	0.4904	0.4157	0.2778	0.0975	-0.0975	-0.2778	-0.4157	-0.4904
3	0.4619	0.1913	-0.1913	-0.4619	-0.4619	-0.1913	0.1913	0.4619
4	0.4157	-0.0975	-0.4904	-0.2778	0.2778	0.4904	0.0975	-0.4157
5	0.3536	-0.3536	-0.3536	0.3536	0.3536	-0.3536	-0.3536	0.3536
6	0.2778	-0.4904	0.0975	0.4157	-0.4157	-0.0975	0.4904	-0.2778
7	0.1913	-0.4619	0.4619	-0.1913	-0.1913	0.4619	-0.4619	0.1913
8	0.0975	-0.2778	0.4157	-0.4904	0.4904	-0.4157	0.2778	-0.0975

Şekil 3.6. AKD matrisi

AKD Katsayılarının Hesaplanması

AKD kendi içerisinde yoğun kosinüs, çarpım ve toplam işlemleri barındırdığından dolayı, bu işlemlerin her bir blok için tekrar tekrar yapılması etkin bir çözüm olmayacaktır. İşlem

yoğunluğunu azaltmak ve işlem hızını arttırmak için Denklem 3.4'teki hesaplama kullanılmaktadır. Bu denklem Şekil 3.5'te bulunan veri matrisi için uygulandığında oluşan AKD katsayı matrisi sonucu Şekil 3.7'de gösterilmiştir.

	1	2	3	4	5	6	7	8
1	305.7500	-3.7956	2.5803	-1.2111	-1.2500	0.6941	-1.6100	0.5195
2	-3.2252	-2.3110	2.2974	-1.0673	0.4645	0.9077	0.2756	-0.7344
3	1.0360	-2.2799	-0.5303	-0.6786	-0.3827	0.4276	-0.9268	-0.2584
4	0.1127	-0.4189	0.1977	-0.8983	0.2364	-0.2164	0.9309	-0.9732
5	0.2500	-0.6043	0.1913	-1.1197	0.2500	0.4030	0.4619	-0.5863
6	-0.6506	0.4126	-0.8838	0.1371	0.2825	-1.2482	-0.2590	0.2276
7	1.1945	0.2975	0.5732	0.5308	-0.9239	0.9103	0.5303	-0.8020
8	-0.5359	-0.3808	-0.4042	0.4458	-0.5230	0.4933	-0.6921	-0.5425

Şekil 3.7. AKD'nin uygulanması sonucunda oluşan katsayı matrisi

Şekil 3.7'de gösterilen matris bir AKD katsayı matrisi olup; matrisin sol üst köşesinde olan ilk katsayı (ilk satır ve ilk sütunda bulunan değer) düşük frekanslı katsayı olup; bu katsayı DC katsayısı olarak ifade edilir. Bu katsayıdan tüm yönlere gidildiğinde (matriste sağa veya aşağı yönde) daha yüksek frekanslı katsayılara ulaşılır. Bu katsayıların her biri AC (63 adet) katsayısı olarak ifade edilir. DC katsayısı blokta bulunan bilginin büyük bir kısmını barındırır.

İnsan gözü düşük frekanslara çok duyarlı olduğu için bir sonraki safha olan niceleme safhasında, bu gerçek göz önünde bulundurularak işlemler gerçekleştirilir [38]. DC katsayısı ve üst sol tarafta bulunan AC katsayıları resmin enerjisinin çoğunu barındırdığı için bu katsayıların muhafaza edilerek; daha az enerji barındıran yüksek frekanslı AC katsayılarının atılması resmin kalitesinde büyük bir değişime neden olmaz [13].

3.2.3. Niceleme

AKD sürecinde oluşturulan katsayı matrisinin niceleme işlemine tabi tutulduğu süreçtir. İnsan gözü, nispeten geniş bir alandaki parlaklıktaki küçük farklılıkları görselleştirmede oldukça iyidir, ancak yüksek frekanslı parlaklık değişiminin kesin gücünü ayırt etmede o kadar iyi değildir [39]. Bu neden yüksek frekanslı bileşenlerin bilgisinin atılmasına imkân sağlar. Bu süreçte insan gözünün algılayamadığı yüksek frekanslı AC katsayılarının elimine edilir. JPEG'nin sıkıştırma işleminin gerçekleştiği en önemli süreçtir.

Bu safhada çeşitli niceleme matrislerinin seçimi ile resim kalitesi ve sıkıştırma oranı değiştirilebilir. Niceleme adımı, 1-100 arasında herhangi bir resim kalitesi seçimi yapma imkânı sunmaktadır. Yüz değeri en yüksek resim kalitesini ifade ederken; bir değeri maksimum sıkıştırma

anlamına gelen en düşük resim kalitesini belirtir. Bu durum gerçekleştirilen uygulamaya ve gereksinimlere göre değişken resim kalitesi veya sıkıştırma oranı sağlar.

Niceleme matrisi 8 X 8 boyutunda bir matris olup; AKD katsayı matrisinin niceleme matrisine bölümünden elde edilen sonuç yine 8 X 8 boyutunda bir matristir. Bu işlem Denklem 3.6'da gösterildiği gibi gerçekleştirilir.

$$K(i,j) = Tamsayıya\ Yuvarla\left(\frac{D(i,j)}{Q(i,j)}\right) \tag{3.6}$$

JPEG için çeşitli sabit niceleme matrisler bulunmaktadır. İstenildiğinde bunlardan biri kullanılabilir. Bu matrislerden en çok yaygın kullanılanı (Q_{50}) şu şekildedir.

$$Q_{50} = \begin{Bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{Bmatrix}$$

Denklem 3.7'deki hesaplama ile standart niceleme matrisinden (Q_{50}) farklı niceleme matrisleri oluşturabiliriz [40].

$$Q_n = \begin{cases} \left(\frac{100-n}{50}\right) Q_{50}, & n > 50 \\ \left(\frac{50}{n}\right) Q_{50}, & n < 50 \end{cases} \tag{3.7}$$

Denklem 3.7'de resmin kalite seviyesini gösteren n değeri 50'den büyük olması durumdan yüksek resim kalitesi ve düşük sıkıştırma oranını sağlarken, 50'den küçük olması durumunda ise yüksek sıkıştırma ve düşük resim kalitesini sağlar [40].

Şekil 3.7'de elde edilen AKD katsayılarının standart niceleme matrisi ile niceleme işlemine tabi tutulması sonucunda oluşan 8 X 8 boyutundaki matris Şekil 3.8'de gösterilmiştir.

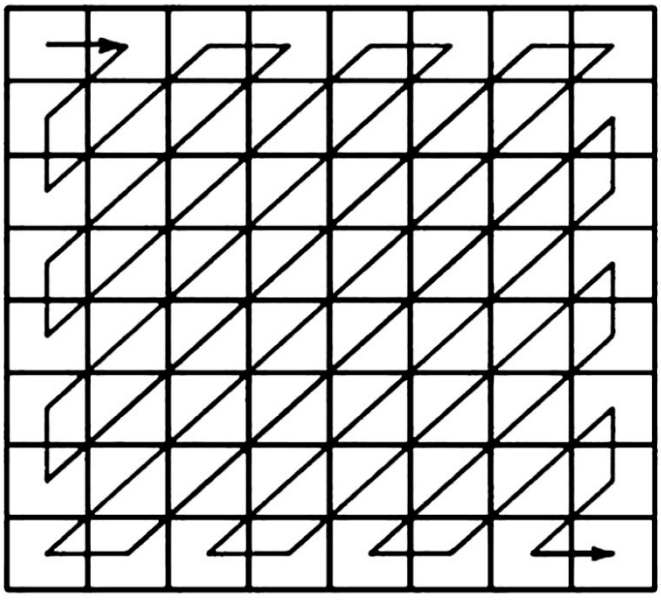
	1	2	3	4	5	6	7	8
1	19	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0

Şekil 3.8. Niceleme işleminin gerçekleştirilmesi

Şekil 3.8'den görüleceği üzere niceleme işleminde yüksek frekanslı AC katsayıları sıfır değerini almıştır. Bütün AC katsayılarının sıfır olduğu durum her zaman gerçekleşmez. Fakat bu safhada yüksek frekanslı AC katsayılarının büyük bir çoğunluğu elemine edilir. Elde edilen bu matris sonraki JPEG süreçlerinde kullanılacaktır.

3.2.4. Zikzak Tarama

Bu model RLE algoritmasının nicelenmiş matris verisini verimli bir şekilde sıkıştırma işlemine tabi tutması için geliştirilmiştir [43,44]. Bu model düşük frekanslı katsayıları gruplamak için kullanılır. Zikzak tarama modeli 8 X 8 boyutundaki bir blok için şekil 3.9'da gösterildiği gibi çalışır.



Şekil 3.9. Zikzak tarama modeli

Sıkıştırılacak olan fotoğrafın bütün veri blokları Şekil 3.9'daki model esas alınarak yeniden düzenlenir. Bu model ile yapılmak istenen temel şey, sıfır değerine sahip AC katsayılarının art arda geleceği bir model oluşturmak ve sonraki aşamalarda oluşturulan bu modelin sıkıştırılmasıdır.

3.2.5. Çalışma Uzunluğu Kodlaması Algoritması

Çalışma uzunluğu kodlaması (RLE), Zikzak tarama evresinde oluşturulan ve çok sayıda ardışık veri tekrarı barındıran bloklar için uygulanan basit kayıpsız bir veri sıkıştırma algoritmasıdır [42]. Bir veri katarında ardı ardına gelen aynı karakterlerin veya değerlerin sayısını bulup; çıktı olarak elde ettiği sayıyı ve değerin kendisini gönderir.

Bu algoritmaya giriş verisi olarak 'aaaaabbbccccccccddddddd' karakter kümesi verildiğini varsayalım. Algoritma karakter kümesinde bulunan bütün değerleri teker teker okuyacak, ardışık gelen aynı karakterlerin sayısını bulup; bunları çıkışa yazacaktır. Örnek olarak verilen karakter kümesinde ardı ardına beş adet 'a', üç adet 'b', dokuz adet 'c' ve yedi adet 'd' karakterlerinden oluşuyor. Algoritma çıktı olarak '5a3b9c7d' karakterini verecektir. Örnek olarak alınan karakter kümesinin uzunluğu 24 olmasına karşın RLE algoritmasının çıktısı sekiz adet karakterden oluşmaktadır.

Bu algoritmanın her türlü veri için uygulanması, her zaman için iyi sonuçlar vermez. Örneğin 'abcd' karakter kümesinin RLE algoritması uygulanması sonucunda oluşan sonuç '1a1b1c1d' olup giriş uzunluğu dört karakter olan bir katar için sonuç olarak üretilen katar sekiz karakter uzunluğundadır.

Çok fazla işlem gerektirmediğinden uygulanması kolay olan bu algoritma, çok sayıda tekrar barındıran veri bloğunda kullanılması verim sağlar [43].

3.2.6. Sıfır Çalışma Uzunluğu Kodlama

Eğer biz dizide en çok tekrar eden karakteri biliniyorsa; bu karakter göz önünde bulundurularak RLE algoritması geliştirilebilir [44]. Zikzak tarama sonucunda oluşan veri dizisinde en çok tekrar eden sıfır değeridir. RLE algoritmasının farklı bir versiyonu olan RLE-0 algoritması; bir veri dizisindeki sıfır değerlerinin sayısını göz önünde bulundurarak sıkıştırma işlemi gerçekleştirir.

Temel mantığı Zikzak taramanın çıkışı olan dizideki, sıfır olmayan AC katsayısından önce ardışık sıfır değerli AC katsayılarını saymaya dayanır [45]. Algoritma sıfırdan farklı bir AC katsayısı buluncaya kadar gelen diziyi okur, bu okuma esnasında gelen sıfır değerlerini sayar. Sıfırdan farklı bir katsayı bulduğunda bu sayma sonucunda elde ettiği sıfırların sayısını ve okuduğu sıfırdan farklı AC katsayısını çıkış olarak yazar. Algoritma blok sonuna ulaşmadan önce sıfırdan farklı bir AC katsayısı ile karşılaşır ve daha önce karşılaştığı sıfır değerli AC katsayılarının sayısı

15 değerine ulaşmış ise çıktıya ‘15’ ve ‘0’ değerleri yazılır. Algoritma 64 katsayıdan oluşan blok dizisinin sonuna ulaştığında ise, saydığı sıfır değerlerinin sayısı yerine özel bir karakter olan ve blok sonunu ifade eden kodunu (‘00’) çıkışa yazar. Algoritma bu kurallar üzerine çalışmasını gerçekleştirir.

Genellikle bir resmin komşu blokları arasında büyük renk ve değer farklılıkları olmaz. Bu sebepten dolayı her bir bloğun DC katsayısının sıkıştırılacak olan dosyada tutulmasının yerine, ilgili bloktan önceki bloğun DC katsayısı ile mevcut bloğun DC katsayısının farkının alınması sonraki aşamalarda oluşacak farklı DC katsayılarının sayısını azaltacaktır. Bu sebepten dolayı bu fark işlemi bu aşamada gerçekleştirilmiştir. DC katsayılarından farklı olarak fark alma işlemi AC katsayıları için gerçekleştirilmez.

3.2.7. Huffman Kodlama

David A. Huffman tarafından 1952 yılında geliştirilen Huffman kodlama, bir dosyada bulunan her bir sembol veya karakter için özel bir kod üretmeye dayanan bir sıkıştırma yöntemidir. Algoritma dosya içerisinde sıkça tekrar eden karakterler için kısa kodlar kullanmaya; dosya içerisinde daha az sayıda bulunan karakterler için ise algoritmanın işleyişine uygun olarak daha uzun kodlar kullanmaya dayanır. Huffman algoritması kayıpsız bir sıkıştırma algoritmasıdır.

Huffman Ağacının Oluşturulması

Algoritma öncelikle dosya içerisinde bulunan farklı karakterlerin bu dosya içerisinde bulunma olasılıklarını hesaplar. Bu olasılıklar göz önünde bulundurularak Huffman ağacını oluşturur. Oluşan ağaca göre her bir karakter için bir kod oluşturur ve bu kodlama esas alınarak dosya tekrar oluşturulur. Huffman kodlamasında yapraktan köke doğru bir ağaç oluşturulur. Algoritmanın işleyiş adımları aşağıda verilen örnekle ifade edilmiştir.

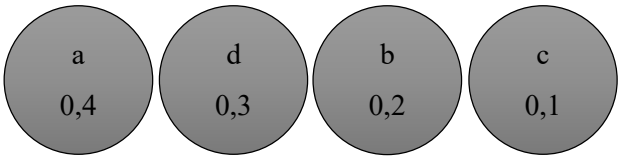
1. Olasılık tablosunun oluşturulması

Sıkıştırmaya tabi tutulacak olan dosyada kullanılan farklı karakterlerin sayısının dört olduğunu ve bu karakterlerin dosya içerisinde kullanılma olasılıklarının ise Tablo 3.1’de verildiği gibi olduğunu varsayalım.

Tablo 3.1. Sembollerin olasılıklarının hesaplanması

Sembol	Sembolün Kullanılma Oranı
a	0.4
d	0.3
b	0.2
c	0.1

Huffman kodlamasında kullanılan tüm sembollerin olasılıklarının toplamının bir değerine eşit olması gerekmektedir. Olasılık değerlerine göre semboller büyükten küçüğe doğru sıralanarak Tablo 3.1 elde edilmiştir. Tablo 3.1'deki değerlere göre oluşturulan Huffman ağacının yaprakları Şekil 3.10'da verilmiştir.



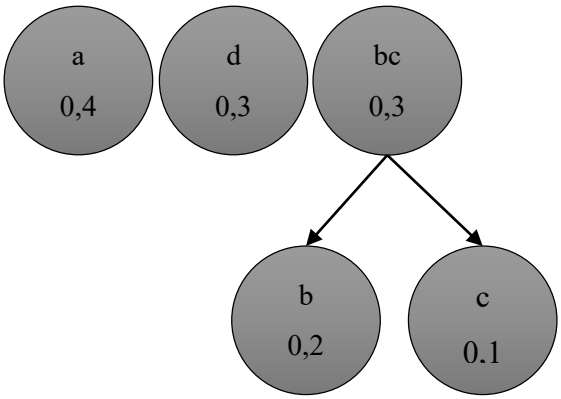
Şekil 3.10. Huffman ağacının düğümlerinin oluşturulması

2. En küçük olasılık değerine sahip iki sembolün olasılıkları toplanarak yeni bir sembol oluşturulur. Oluşturulan yeni sembolde göz önünde bulundurularak ilk aşamadaki gibi sembollerin olasılıklarına göre bir tablo düzenlenir. Tablo 3.2'de bu işlemin sonucu gösterilmiştir.

Tablo 3.2. Olasılık tablosunun güncellenmesi

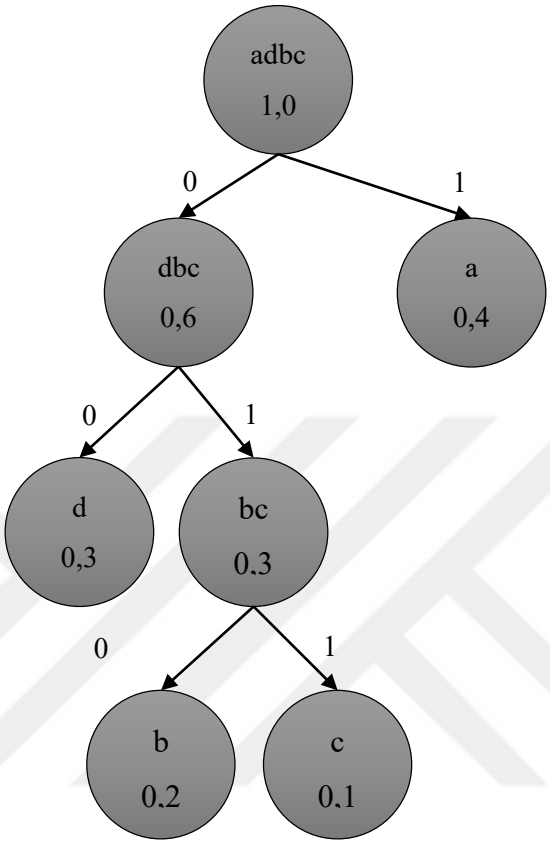
Sembol	Sembolün Kullanılma Oranı
a	0.4
d	0.3
bc	0.3

Tablo 3.2'de verilen değerlere göre Huffman ağacı tekrar oluşturulur. Oluşturulan Huffman ağacı Şekil 3.11'de gösterilmiştir.



Şekil 3.11. Huffman ağacının yeniden oluşturulması

3. Tek bir düğüm elde edilinceye kadar 2. adımdaki işlemler tekrarlanır. İşlemler sonucunda elde edilen Huffman ağacı Şekil 3.12'de gösterilmiştir.



Şekil 3.12. Huffman Ağacının oluşturulması

Şekil 3.12'deki Huffman ağacına göre her bir karakter için belirlenen kod Tablo 3.3'te verilmiştir.

Tablo 3.3. Hufman ağacına göre sembollerin kodları

Sembol	Karakter Kodu
a	1
b	010
c	011
d	00

Huffman algoritması her bir karakter için kod belirleme işlemini gerçekleştirdikten sonra sıkıştırılacak veriyi barındıran dosyadaki sembollere göre Tablo 3.3'te karşılık gelen ilgili karakterler yazılarak çıkış dosyası hazırlanır.

Huffman algoritması, içerisinde az sayıda farklı karakter barındıran yüksek boyutlu dosyalara uygulanması sıkıştırma yönünden avantaj sağlayabilir [46]. Algoritmada oluşturulan kodlama tablosu sıkıştırılmış veriye eklenmelidir. Çünkü bu kodlama tablosu göz önünde bulundurularak dosya tekrardan oluşturulur.

JPEG dosyasının başlık kısmında, kullanılan Huffman ağacı veya ağaçlarının tablo halinde eklendiği bir kısım vardır. Resmin çözülmesi esnasında çözücü JPEG resminin başlık kısmında bulunan etiketlerden Huffman tablosunu çıkarıp; bu tabloya göre kod çözme işlemini gerçekleştirir.

JPEG resim sıkıştırma modelinde Huffman kodlaması aşamasında birden fazla Huffman tablosu kullanılabilir. DC katsayıları ve AC katsayıları için farklı Huffman tablosu kullanabilir.

3.3. JPEG Enerji Modeli

Bu bölümde JPEG sıkıştırma aşamalarının gerçekleştiği kaynak düğümde meydana gelen işlem sayıları verilmiştir. Bir resmin sıkıştırılması sürecinde kullanılan enerji miktarı E_p ile ifade edilirse, sıkıştırma sürecinde kullanılan enerji miktarı şöyle ifade edilebilir:

$$E_p = E_{AO} + E_{AKD} + E_N + E_Z + E_{RLE} \tag{3.8}$$

Burada E_{AO}; aralık ortası, E_{AKD} ayrık kosinüs dönüşümü, E_N niceleme süreci, E_Z zikzak tarama süreci ve E_{RLE} çalışma uzunluğu kodlaması enerji miktarını ifade etmektedir. Enerji hesaplama yönteminde işlemlerin gerçekleştirilmesi süresince yapılan hafıza erişimlerinin tüketilen enerji miktarına etkisi hesaplanmamıştır.

3.3.1. Aralık Ortası Enerji Modeli

Aralık ortası işleminde 8 X 8 blok verisi için, her piksel değerinden 128 değeri çıkarılarak yeni blok verisi elde edilmekteydi. Bu bağlamda N X N boyutundaki bir resim için gerçekleştirilecek çıkarma işlemlerinin sayısı Denklem 3.9'da gösterilmiştir.

$$E_{ao} = N^2 e_{\text{çıkarma}} \tag{3.9}$$

Denklem 3.9'da gösterilen $e_{\text{çıkarma}}$ ile kaynak düğümün bir adet çıkarma işlemini gerçekleştirmesinde tükettiği enerji miktarı ifade edilmiştir.

3.3.2. AKD Enerji Modeli

N x N boyutundaki bir resmin k x k boyutundaki küçük bloklara bölünmesi sonucunda oluşan blok sayısı $\left(\frac{N}{k}\right)^2$ değerine eşittir. AKD her bir blok için gerçekleştirilir. Denklem 3.4'ün her k x k matris sonucu hesaplamada k^2 katsayı içerir [11]. Aynı denklemde her bir katsayı için k adet çarpma

işlemi ve (k-1) adet toplama işlemi yapılması gerekmektedir. Bu sebepten dolayı iki matrisin çarpımı için harcanan enerji miktarı $2k^2\left(ke_{çarpma} + (k-1)e_{toplama}\right)$ olur. AKD sürecinde harcanan toplam enerji Denklem 3.10'da gösterildiği gibi hesaplanır.

$$E_{AKD} = \left(\frac{N}{k}\right)^2 e_{AKD} = \left(\frac{N}{k}\right)^2 2k^2\left(ke_{çarpma} + (k-1)e_{toplama}\right) \quad (3.10)$$

Denklem 3.10'da bulunan $e_{çarpma}$ düğüm tarafından bir çarpma işlemini gerçekleştirilmesi için harcanan enerji miktarını temsil ederken; $e_{toplama}$ düğümde bir toplama işleminin gerçekleştirilmesi için harcanan enerji miktarını temsil eder [11].

3.3.3. Niceleme Enerji Modeli

Niceleme safhasında Denklem 3.6'da gösterildiği gibi k x k boyutuna sahip bir blokta bulunan her bir katsayı için bir adet bölme işlemi ve bir adet yuvarlama işlemi gerçekleştirilmektedir [11]. Bir blok için niceleme safhasında k^2 adet bölme işlemi ve k^2 adet yuvarlama işlemi gerçekleştirilir. Bu bağlamda N X N boyutundaki bir resmin k x k boyutundaki bloklara bölünmesinden elde edilen blok sayısı $\left(\frac{N}{k}\right)^2$ olup; niceleme aşamasında harcanan enerji miktarı Denklem 3.11'de gösterildiği gibi hesaplanır.

$$E_N = \left(\frac{N}{k}\right)^2 e_{nic} = \left(\frac{N}{k}\right)^2 k^2\left(e_{bölme} + e_{yuvarlama}\right) \quad (3.11)$$

Denklemde bulunan $e_{bölme}$ bir bölme işleminin kaynak düğüm tarafından gerçekleştirilmesinde tüketilen enerji miktarı olup; $e_{yuvarlama}$ bir yuvarlama işleminde harcanan enerji miktarını ifade eder [11].

3.3.4. Zikzak Enerji Modeli

Bu aşamada k x k boyutundaki blokta bulunan AC katsayılarının kaydırılması işlemi gerçekleşmektedir [11]. Her bir blokta bir adet DC katsayısı olup; bloktaki AC katsayılarının sayısı (k^2-1) değerine eşittir. N X N boyutundaki bir resmin k x k boyutundaki bloklara bölünmesinden elde edilen blok sayısı $\left(\frac{N}{k}\right)^2$ olup; zikzak aşamasında harcanan enerji miktarı Denklem 3.12'de gösterildiği gibi hesaplanır.

$$E_Z = \left(\frac{N}{k}\right)^2 e_z = \left(\frac{N}{k}\right)^2 (k^2 - 1)e_{kaydırma} \quad (3.12)$$

Denklem 3.12'de geçen $e_{kaydırma}$ kaynak düğümde bir kaydırma işleminde tüketilen enerji miktarını temsil eder.

3.3.5. Çalışma Uzunluğu Kodlaması Enerji Modeli

Çalışma uzunluğu kodlaması bir önceki safhada (zikzak tarama) üretilen (k^2-1) AC katsayısının yeniden düzenlenmesini sağlar. Çalışma uzunluğu kodlamasının enerji miktarı [10]'da şöyle hesaplanır:

- e_z: Bir AC katsayısının sıfır olup olmadığını kontrol etmek için harcanan enerji miktarıdır.
- e_{yaz}: Bir (r,X) değerini yazmak için harcanan enerji miktarıdır. X değeri sıfırdan farklı bir AC katsayısı olup; r bu AC katsayısından önceki ardışık sıfır sayısıdır.
- $e_{arttır}$: Bir sıfır dizisindeki sıfır değerlerini sayan sayacın değerinin bir arttırılması için harcanan enerji miktarıdır.
- $e_{sıfırla}$: Sıfır değerlerini sayan sayacın sıfırlanması için harcanan enerji miktarıdır.
- $N_{0_{dizi}}$: Bir k x k boyutuna sahip blokta bulunan sıfır dizilerinin sayısıdır. Bu sıfır dizilerini 0dizi [0], 0dizi [1], 0dizi [2], … 0dizi [$N_{0_{dizi}}$] ile gösterilir.
- uzunluk(i) : Bir 0dizi[i] dizisindeki sıfır sayısını gösterir.

Bir blokta bulunan toplam sıfır sayısı Denklem 3.13'te gösterildiği gibi hesaplanır [10].

$$N_0 = \sum_{i=1}^{N_{0_{dizi}}} uzunluk(i) \tag{3.13}$$

Toplam enerji şöyle hesaplanabilir [10] :

$(k^2 - 1)$ adet AC katsayısının kontrol edilmesinde harcanan toplam enerji miktarı $(k^2 - 1)e_z$'dir. $\left((k^2 - 1) - \left(N_0 + N_{0_{dizi}}\right)\right)$ Değeri öncesinde sıfır dizisi bulunmayan sıfırdan farklı AC katsayısının değerini verir ve bu kategorideki her bir X katsayısı (0,X) yazmayı gerektirir ki; bu durumda harcanan enerji miktarı $\left((k^2 - 1) - \left(N_0 + N_{0_{dizi}}\right)\right) e_{yaz}$ ile hesaplanır. Öncesinde sıfır dizisi bulunan sıfırdan farklı her bir AC katsayısı için algoritmanın adımlarından birisi olan ayrıştırma safhası Denklem 3.14'te verilen biçimde düşünülür.

$$p_i = uzunluk(i)\ bölüm\ 16 \text{ ve } q_i = uzunluk(i)\ mod\ 16 \tag{3.14}$$

Ayrıştırmadan sonra $(p_i + 1)0_{dizi}$ elde edilir [10]. Her bir sıfır dizisi için sayacı sıfırlamak ve elde edilen (r,X) değerini yazmak için harcanan enerji miktarı Denklem 3.15'te verilmiştir.

$$(p_i + 1)\left(e_{sıfırla} + e_{yaz}\right) \tag{3.15}$$

Her bir ilk p_i 0_{dizi} (uzunluğu 16 olan) sayacı 15 defa arttırmayı ve son 0_{dizi} (uzunluğu q_i) sayacı $q_i - 1$ defa arttırmayı gerektirir, toplam enerji miktarı Denklem 3.16'da gösterildiği gibi hesaplanır [10] :

$$(15p_i + q_i - 1)e_{arttır} \tag{3.16}$$

0dizi[i] ayrışma süreci için $e_{0_{dizi}}(i)$ Denklem 3.17'de gösterildiği gibi denklem haline getirilebilir [10].

$$e_{0_{dizi}}(i) = \begin{cases} (p_i + 1)\left(e_{yaz} + e_{sıfırla}\right) + (15p_i + q_i - 1)e_{arttır}, i < N_{0_{dizi}} \; V \; (SonBileşen \neq 0) \\ e_{yaz} \qquad\qquad , diğer \; durumlarda \end{cases} \tag{3.17}$$

Bir k x k boyutundaki blok için harcanan toplam enerji miktarı Denklem 3.18'de gösterilmiştir [10].

$$e_{rle} = (k^2 - 1)e_z + \left((k^2 - 1) - \left(N_0 + N_{0_{dizi}}\right)\right)e_{yaz} + \sum_{i=1}^{N_{0_{dizi}}} e_{0_{dizi}}(i) \tag{3.18}$$

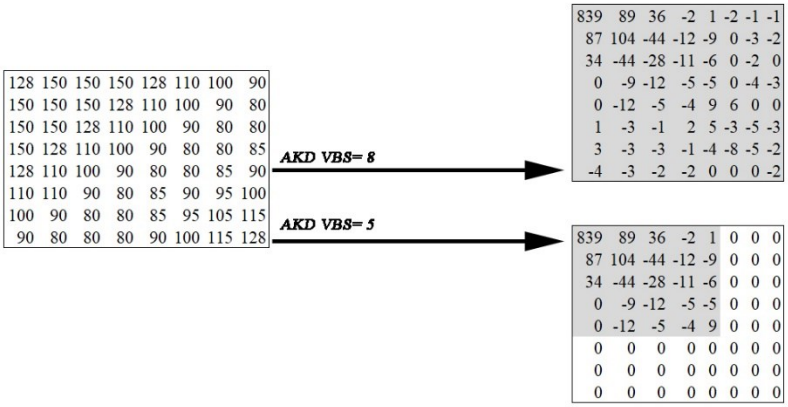
Bir N X N boyutundaki resim için RLE safhasında kullanılan enerji miktarı Denklem 3.19'da gösterildiği gibi hesaplanır [10].

$$E_{rle} = \sum_{j=1}^{\left(\frac{N}{k}\right)^2} e_{rle\,j} \tag{3.19}$$

4. KMAA İÇİN JPEG UYARLAMALARININ İNCELENMESİ

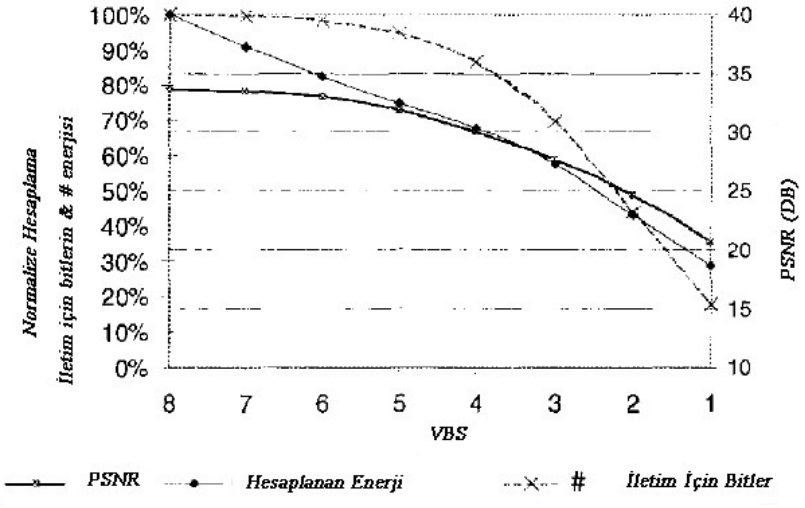
Bölüm 3'te JPEG süreçleri hakkında bilgiler verilmişti. Bu süreçler içerisinde özellikle AKD sürecinde işlem yoğunluklarından dolayı JPEG'in, KMAA gibi güç kısıtlamalı uygulamalarda kullanılması çok verimli bir sonuç oluşturmaz. Bu sebepten dolayı JPEG'in işlem yoğunluğunu azaltan çeşitli uygulamalar literatürde önerilmiştir. Bu tezde önerilen D-LEICA algoritmasının temelinde bu uygulamalar bulunmaktadır.

Taylor C. N. ve Dey S. [9]'da KMAA iletişiminde JPEG'in niceleme ve AKD süreçleri için kullanılan parametreleri değiştirmek suretiyle bir model önerdiler. AKD sürecinde 8 x 8 boyutundaki bir blok için işlemler gerçekleştirilir. Bu süreçte elde edilen katsayıların büyük bir çoğunluğu niceleme safhasında sıfır olmaktadır. Bu bağlamda bütün katsayıların hesaplanması işlem ve enerji açısından ekstra bir yük anlamı ihtiva eder. Taylor C. N. ve Dey [9] VBS (Sanal Blok Boyutu) olarak nitelendirdikleri ve AKD sürecinde kullanılan bloğun bir alt bloğu niteliğinde bir yapı kullandılar. Tüm bloğun AKD katsayılarını hesaplamak yerine VBS bloğundaki AKD katsayıları hesaplamayı önerdiler. Bu yöntem aynı zamanda niceleme safhasında da işlem yükünü azalttığı için tüketilen enerji miktarını azaltır. Ayrıca kodlama safhasında (Huffman veya aritmetik kodlama) düşük bitle ifade edilen sıfır değerlerinin sayısını arttırdığı için veri iletiminde çeşitli avantajlar sağlar. Şekil 4.1'de VBS değerleri 8 ve 5 olan bloklar için AKD girişleri gösterilmiştir.



Şekil 4.1. Farklı sanal blok boyutları için AKD girişleri [9]

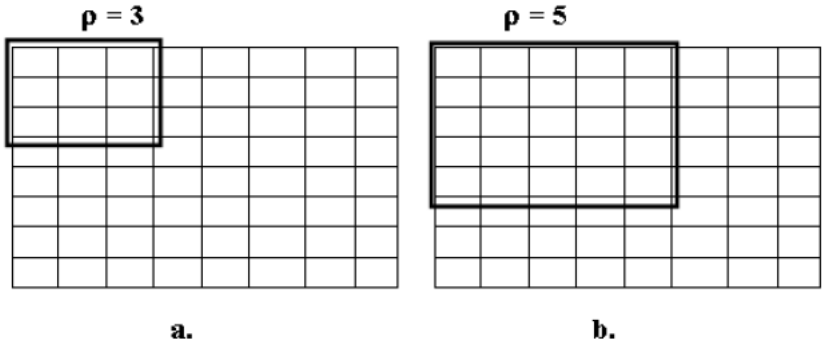
Şekil 4.2 VBS değerlerinin azalmasıyla iletilecek veri boyutunun miktarı, hesaplama için harcanan enerji miktarı ve resim kalitesindeki ilişkiyi göstermektedir.



Şekil 4.2. VBS ile iletilen veri boyutu, hesaplama için harcanan enerji miktarı ve resim kalitesi arasındaki ilişki [9]

4.1. S-JPEG

Bir AKD bloğunun enerjisinin büyük bir çoğunluğu DC katsayısı ve düşük frekanslı AC katsayılarında bulunur. Bu gerçeği temel alan Mammeri A. vd. [10] S-JPEG (Kare-JPEG) olarak nitelendirdikleri bir model tanımladılar. Bu model VBS'e benzer olup; k x k boyutunda olan her bir bloğunun sol üst köşesinde bir kare (ρ uzunluğunda) üzerinde AKD işlemlerinin gerçekleştirilmesi esasına dayanır. JPEG'nin bir uyarlama modeli olan S-JPEG ρ değeri ile belirlenen kare içerisine düşen katsayılar için işlem gerçekleştirdiği için enerji verimliliği sağlar. Burada kullanılan ρ değeri ile resim kalitesi arasında doğru orantılı bir ilişki vardır. Eğer ρ değeri artar ise resim kalitesi ve iletilecek veri boyutu da artar. Buna bağlı olarak harcanan enerji miktarı da fazladır. Şekil 4.3'te farklı ρ değerine göre kare seçimi gösterilmiştir.



Şekil 4.3. ρ değerinin seçimi

4.1.1. S-JPEG Enerji Modeli

Bir k x k boyutundaki veri bloğunda k^2 adet katsayı için AKD işlemleri gerçekleştirilir. S-JPEG modelinde kullanılan AKD bloğu ρ x ρ boyutuna sahip olup; ρ^2 adet katsayı için işlemler gerçekleştirilir. Burada $k^2 > \rho^2$ ilişkisi göz önünde bulundurularak, S-JPEG modelinin enerji verimliliğinin JPEG'e göre daha iyi olduğu söylenebilir [10]. S-JPEG enerji tüketim modeli JPEG enerji modeline benzerdir. S-JPEG modelinde harcanan enerji miktarı $E^{\Diamond}$ ile gösterilirse; tüketilen toplam enerji Denklem 4.1'de gösterildiği gibi hesaplanır.

$$E^{\Diamond} = E_{AKD}^{\Diamond} + E_{N}^{\Diamond} + E_{Z}^{\Diamond} + E_{RLE}^{\Diamond} \tag{4.1}$$

$E_{AKD}^{\Diamond}$, $E_{N}^{\Diamond}$, $E_{Z}^{\Diamond}$ ve $E_{RLE}^{\Diamond}$ sırasıyla AKD, niceleme, zikzak tarama ve çalışma uzunluğu kodlaması safhasında S-JPEG modelinde tüketilen enerji miktarını gösterir. JPEG enerji modeline benzer bir şekilde S-JPEG içinde bir enerji modeli oluşturulabilir.

[10]'da ana istasyonda uygun görülebilecek bir resim kalitesi elde etmek için ρ değeri dört olarak alınmış, tüketilen enerji miktarı 3.74 mJ (mili Joule) olarak ölçülmüş; sekiz değeri için tüketilen enerji miktarı 10,47 mJ olarak ölçülmüştür. Elde edilen bulgular S-JPEG'nin enerji konusunda JPEG'den daha iyi olduğu sonucuna varılmıştır. Ayrıca kümelenmiş bir KMAA için farklı ρ değerleri için (iki, dört, altı) S-JPEG ve JPEG modellerine göre yaşam süreleri test edilmiş ve çeşitli ρ değerleri için S-JPEG'nin daha iyi olduğu sonucuna varılmıştır. Yapılan testlere göre JPEG ile 34 çevrim yapılırken; ρ=6 için 66 çevrim, ρ=4 için 101 çevrim, ρ=1 için 120 çevrim yapılmasına karşın 15 adet düğümden oluşan kümede hala 10 adet düğümün enerjisinin olduğu tespit edilmiştir.

4.2. LEICA

LEICA (Düşük Enerji Resim Sıkıştırma Algoritması) referans olarak alınan bir arka plan resmine göre (kameranın sürekli olarak izlediği sabit alan) algılayıcının herhangi bir nesne algıladıktan sonra çektiği fotoğraf (ön plan resmi) arasındaki değişimleri göz önünde bulundurarak; sıkıştırılacak resim için önemli alan ve arka plan olmak üzere sınıflandırma gerçekleştirir.

Bunun için ilk olarak arka plan ve ön plan resimleri 8 x 8 matris boyutundaki bloklara bölünür. Arka plan (b) ve ön plan (f) resimlerinin ilgili blokları arasındaki piksel değerlerinin farklarının ortalaması, Denklem 4.2'de gösterildiği gibi hesaplanır.

$$D_{m,n} = \frac{1}{64} \sum_{i=1}^{8} \sum_{j=1}^{8} |f(8m+i, 8n+j) - b(8m+i, 8n+j)| \tag{4.2}$$

Burada kullanılan m ve n değerleri sırasıyla resmin yatayda ve düşeyde kaç adet bloklara bölündüğünün sayısıdır. i ve j değerleri blokların ilgili satır ve sütun indeks değerleridir. Her bir blok için $D_{m,n}$ değeri hesaplandıktan sonra, önceden belirlenmiş bir eşik değerine (D_0) göre

sınıflandırma işlemi gerçekleştirilir. D_0 sıkıştırma oranının belirlenmesinde önemli rol oynar [13]. Şekil 4.4'te yukarıda anlatılan sınıflandırma işlemine göre belirlenmiş resimler görülmektedir.

a) Arka plan **b) Ön plan** **c) Önemli alan**

Şekil 4.4. LEICA'nın temel işleyiş biçimi [53]

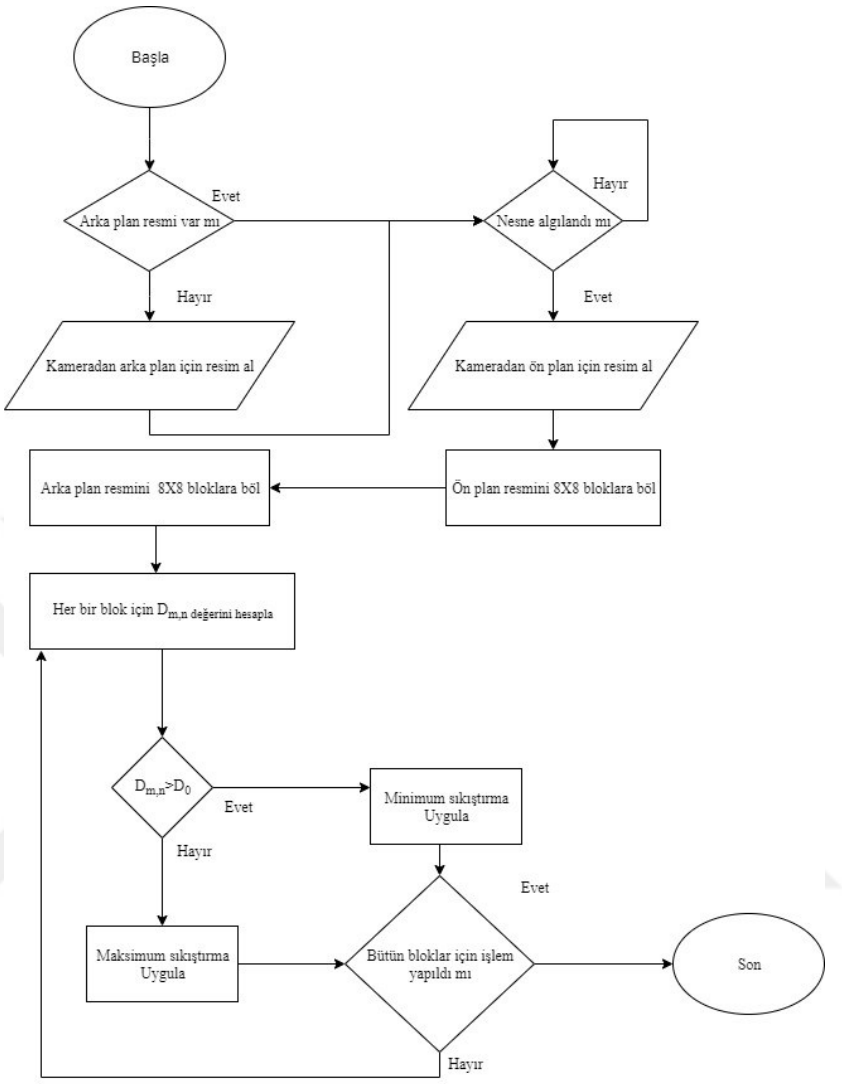
$D_{m,n}$ değeri hesaplandıktan sonra Denklem 4.3'te gösterilen hesaplamayla AKD sürecinde hesaplanacak katsayıların sayısı (ρ değeri) belirlenir [13].

$$\rho_{m,n} = \begin{cases} \rho_{maks}, & D_{m,n} > D_0 \\ \rho_{min}, & diğer \end{cases} \tag{4.3}$$

Denklem 4.2 ve Denklem 4.3'teki hesaplamalar resimde bulunan her bir 8 X 8 boyutundaki blok için gerçekleştirilir ve elde edilen ρ değerine göre o bloktaki veriler minimum veya maksimum düzeyde sıkıştırma işlemine tabi tutulur.

Sınıflandırma işlemi gerçekleştirildikten sonra eğer bir alan ilgilenilen bölge olarak sınıflandırılmış ise; bu durumda S-JPEG modeline benzer bir şekilde bir ρ değeri kullanılır. Kullanılan bu değer ρ_{maks} ile ifade edilir. Arka plan olarak sınıflandırılan alan için ise bir ρ_{min} değeri kullanılır. Kullanılan ρ değerine göre resmin ilgili alanı sıkıştırılır. Burada kullanılan ρ_{maks} ve ρ_{min} değerleri uygulamaya veya tercihe bağlı olarak değişkenlik gösterebilir. Kullanılacak ρ değeri bu şekilde belirlendikten sonra LEICA algoritması, JPEG yöntemiyle aynı şekilde diğer süreçleri gerçekleştirir. JPEG süreçlerinde kullanılan blok boyutu 8 X 8 olduğu için LEICA algoritmasında kullanılabilecek ρ_{maks} değeri en fazlan sekiz olabilecekken ve ρ_{min} değeri de en az bir olabilir.

Yöntem ilgilenilen alanda resmin kalitesini garanti etmek için minimum sıkıştırma; arka plan olarak sınıflandırdığı alanda ise maksimum sıkıştırma esasına göre çalışır. Arka plan olarak belirtilen alanda herhangi önemli bir olay veya değişiklik gözlemlenmediği için bu bölge önemsiz olarak nitelendirilir ve bu sebepten dolayı maksimum sıkıştırmaya tabi tutulur. Önemli olarak nitelenen alanda ise arka plan resmine göre değişimler tespit edildiğinden dolayı burada minimum sıkıştırma gerçekleştirilir. Şekil 4.5'te bu tez çalışmasında kullanılmak üzere MATLAB ortamında hazırlanan LEICA uygulamasına ait akış şeması verilmiştir.



. **Şekil 4.5.** LEICA algoritmasının akış şeması

Şekil 4.5'te verilen akış şemasında LEICA algoritması her bir blok için $D_{m,n}$ değeri hesaplandıktan sonra, uygulamaca kullanılan bir eşik değerine göre (D_0) bu değeri kıyaslayarak; ilgili veri bloğunun minimum sıkıştırmaya mı (ρ_{maks}) yoksa maksimum sıkıştırmaya (ρ_{min}) tabi olacağına karar verir. Karar sürecinden sonra gerçekleştirilen işlemler JPEG algoritmasında kullanılan süreçlerdir.

4.2.1. LEICA Enerji Modeli

LEICA'nın enerji modelinin hesaplanma biçimi S-JPEG enerji modeline benzerdir. Eğer bir blok ilgilenilen alan olarak nitelenmiş ve ρ_{maks} alınmış ise hesaplama bunu temel alarak yapılır. Diğer durumlarda ρ_{min} esas alınarak hesaplama yapılır. Kullanılacak ρ değeri eğer ρ_{maks} ise bu değer k değerinden küçük veya eşit olacaktır [13]. Diğer durumda kullanılacak ρ değeri ρ_{min} olup; bu

değer k değerinden küçük olacağı için, enerji verimliliği açısından LEICA'nın kullanımı JPEG'e göre avantajlı olacaktır.

[13]'te yapılan deneylerde LEICA'nın sıkıştırma oranını JPEG ve S-JPEG' göre arttırdığı, LEICA ile gerçekleştirilen uygulamalarda tüm ağın ömrünün bu uygulamalara göre daha fazla olduğu sonucuna varılmıştır.

5. VERİ ŞİFRELEME

Kablosuz algılayıcı ağlarda ilgilenilen alandan toplanan veri uygulama alanına bağlı olarak çok önem taşıyabilmektedir. Elde edilen önemli verinin yetkisiz kişiler tarafından erişilmesini ve iletilecek olan verinin değiştirilmesini engellemek amacıyla çeşitli güvenlik önlemleri alınmalıdır. Aksi takdirde kablosuz algılayıcı düğüm tarafından elde edilen bu bilginin iletimi esnasında istenmeyen kişilerce elde edilmesi ve hatta bu verinin bu kişilerce değiştirilerek merkez istasyona iletimi sağlanabilir.

Verinin gizliğini ve bütünlüğünün temin edilebilmesi için çeşitli şifreleme yöntemleri geliştirilmiştir. Bu yöntemler kullanılmak suretiyle veri şifrelenerek; ilgili yere verinin güvenli bir şekilde iletilmesi sağlanmış ve alıcı tarafta deşifreleme ile asıl veriye ulaşılması amaçlanmıştır.

Bilginin sadece istenilen kişilerce erişilmesini ve işlenmesini amaçlayan Kriptografi bilimi; çeşitli kodların kullanılması ile bilginin korunmasıdır [47]. Kriptografi bilimin başlangıcı binlerce yıl öncesine dayanır. Özellikle 20. yy. daki teknolojik gelişmeler ile birlikte daha karmaşık ve verimli araçlar sağlandı.

5.1. Kriptografi Tarihi

İnsanlık tarihi boyunca gizli ve önem arz eden bilgilerin güvenli bir şekilde aktarılması için çeşitli yöntemler geliştirilmiştir. Tarihte bilinen en eski kriptografi örneği yaklaşık 4000 yıl öncesinde bir mezar duvarında bulunan yazıtların anlamlarını anlaşılmaz hale getirmek için kullanılan alışılmadık hiyerogliflerde bulunur [48]. M.Ö. 5. yy. gizli mesaj alıp göndermek için Spartalı'lar tarafından bir şifreleme yöntemi geliştirildi. Bu yöntem gönderici ve alıcı tarafta bulunan bir silindir üzerine sarılı olan bir parşömen veya deri üzerine mesajın yazılması temeline dayanıyordu. Mesaj içeren parşömen silindirden ayrıldığında, mesaj anlamsız harflerden meydana gelmekteydi. Alıcı tarafta tekrar mesaj silindire sarılmak suretiyle anlamlı hale gelmekteydi.

Sezar şifreleme olarak adlandırılan yöntem ise; gönderici ve alıcı tarafından bilinen bir anahtar değerine göre gönderici tarafta mesaj içerisindeki harflerin bu anahtar değerine göre ileri götürülmesi, alıcı tarafta ise alınan mesaj içerisindeki her bir harfin bu anahtar değerine göre geri getirilmesi esasına dayanır. Örnek olarak anahtar değerinin üç olarak alıcı ve gönderici tarafta kabul edildiği varsayarsak; alıcı tarafta kullanılacak her bir b harfi alfabede (Türk alfabesine göre) üç karakter ileri atılacak ve d değerini alacaktır. Mesaj içerisindeki bütün harfler için bu işlemler gerçekleştirilir ve şifreli metin elde edilir. Alıcı tarafta okunan d harfi için anahtar değeri göz önünde bulundurularak alfabede geri atlanır ve b değeri elde edilir. Deşifreleme bütün metindeki harfler için bu şekilde gerçekleştirilerek düz metin elde edilir.

Sezar şifrelemeden farklı olarak birden fazla alfabe kullanan Vigenere şifrelemede; bir anahtar seçilir. Eğer seçilen anahtarın uzunluğu gönderilecek mesajdan daha kısa ise; anahtar tekrarlı bir şekilde art arda gelecek şekilde eklenerek mesaj boyutuna getirilir. Şifrelenecek mesajın ilgili karakteri ile anahtarın karşılık gelen karakterin kullanıldıkları alfabedeki sıra numaraları toplanarak (Örneğin A harfi 1, B harfi 2 vb.) alfabedeki harf sayısına göre mod almak (Türkçede 29 harf olduğu için mod 29 alınır) suretiyle şifreli karakter elde edilir.

Alman mühendis Arthur Scherbius tarafından icat edilen Enigma makinesi ikinci dünya savaşında Alman askeri güçleri tarafından yoğun bir şekilde kullanıldı [49]. İçerisinde üç veya daha fazla pervane bulunduran Enigma; klavyesinde metin yazılırken pervaneler farklı oranlarda döner ve şifreli çıktıyı oluşturur. Alman askeri güçleri tarafından şifreli bir şekilde iletişim sağlanabilmesi için savaş boyunca kullanılmıştır. Enigma'nın kullandığı şifreleme sistemi günlük ayarlamalar ile değiştiği için bu karmaşık şifrenin insan tarafından çözülmesini olanaksız kılıyordu [50]. Alan TURING tarafından geliştirilen Bombe makinesi Enigma şifresini kırdı. Bu gelişme hem savaşın daha erken sonlanmasını sağladı hem de modern bilgisayar bilimlerinin temelini oluşturdu.

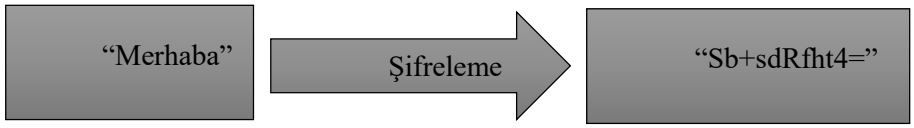
1977 de Ulusal Standartlar Bürosu devrim niteliğinde olan veri şifreleme standardını (DES) ortaya çıkardı [51]. DES ilk evrensel şifreleme standardıydı. Gelişen teknolojiyle birlikte daha güçlü bilgisayarlar üretildi ve DES şifreleme de kırıldı.

NIST tarafından 2000 yılında DES şifrelemenin yerini alması için Joan Daemen ve Vincent Rijmen tarafından tasarlanan, Rijndael algoritmasının Gelişmiş Şifreleme Standardı (AES) olarak kullanılacağını ilan etmiştir [52].

5.2. Veri Şifreleme

Veri şifreleme; verinin sadece bir gizli anahtara sahip kişiler tarafından erişilmesini sağlayan, verinin farklı bir formata dönüşmesini sağlayan bir yöntemdir [53]. Şifreleme, düz veriyi şifreli veri haline dönüştürme işlemidir. Şifreleme [54]'te: Eski kriptografik yöntemlerin farklı bir biçimi olarak tanımlanmıştır. Şifrelenmemiş olan veriye düz veri denir.

Şekil 5.1'de bir şifreleme algoritmasının giriş ve çıkışlarının nasıl olabileceği gösterilmiştir.



Şekil 5.1. Örnek bir şifreleme

Şifreleme sürecinin sonucunda oluşan şifreli veriye bakıldığında oluşan verinin rastgele oluşmuş olduğu gibi görünmelidir. Şifreleme sürecinin tersine uygulanabilir olması gerekmektedir.

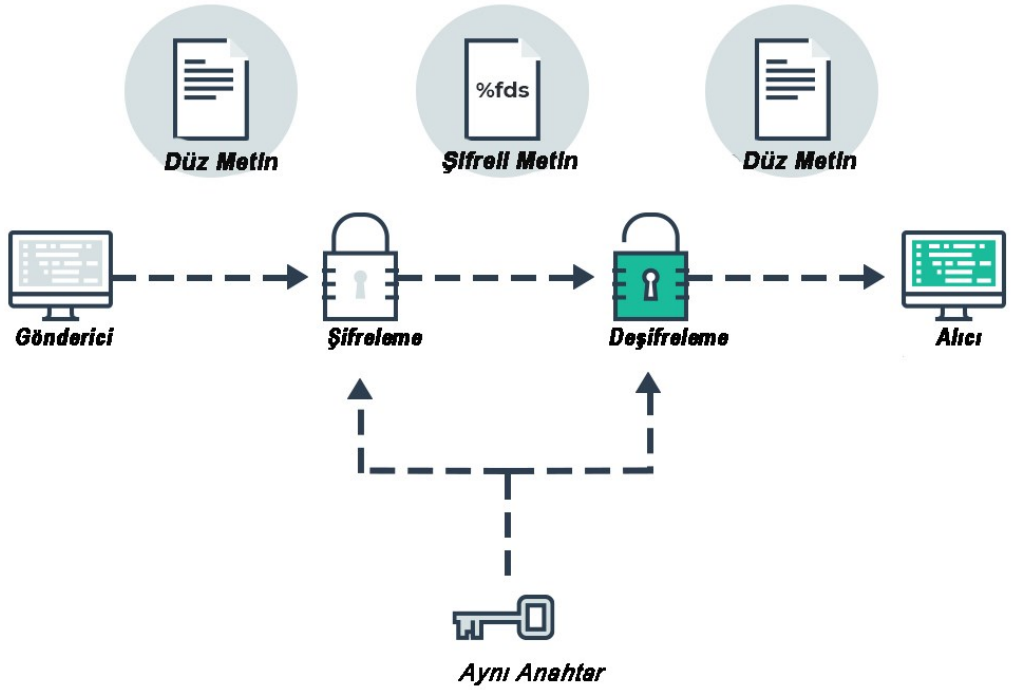
Tersine uygulanamayan bir şifreleme sürecinde düz (şifrelenmeden önceki) veriye ulaşılamaz. Bu bir uygulama için istenmeyen bir durumdur. Belirli bir mantık ve işlem sırasıyla gerçekleştirilen şifreleme işlemi, anahtara sahip bir kullanıcı tarafında şifreleme sürecinin tersi bir süreç işletilerek düz veriye ulaşılır.

5.3. Şifreleme Türleri

İki tür şifreleme yöntemi bulunmaktadır: Simetrik şifreleme ve asimetrik şifreleme. Simetrik şifrelemede alıcı ve gönderici bir tek anahtar kullanarak şifreleme ve deşifreleme işlemlerini gerçekleştirirken; asimetrik şifrelemede bu işlemleri gerçekleştirmek için iki adet anahtar bulunmaktadır.

5.3.1. Simetrik Şifreleme

Alıcı ve göndericinin aynı anahtar ile veriyi şifreleyip ve deşifre ettiği yöntemlerdir. AES, DES, 3DES, RC4, RC6, Serpent ve Blowfish simetrik şifreleme algoritmalarına örnektir. Simetrik şifreleme basamakları Şekil 5.2'de gösterilmiştir.



Şekil 5.2. Simetrik şifreleme ve deşifreleme süreçleri [54].

Simetrik şifrelemenin avantajları aşağıda sıralanmıştır [55]:

- Gerçekleştirilme süresi bakımından hızlıdır.
- Donanımla beraber kullanılabilirler.
- Kullanılan anahtarların boyutları kısadır.
- Kullanımları kolaydır.

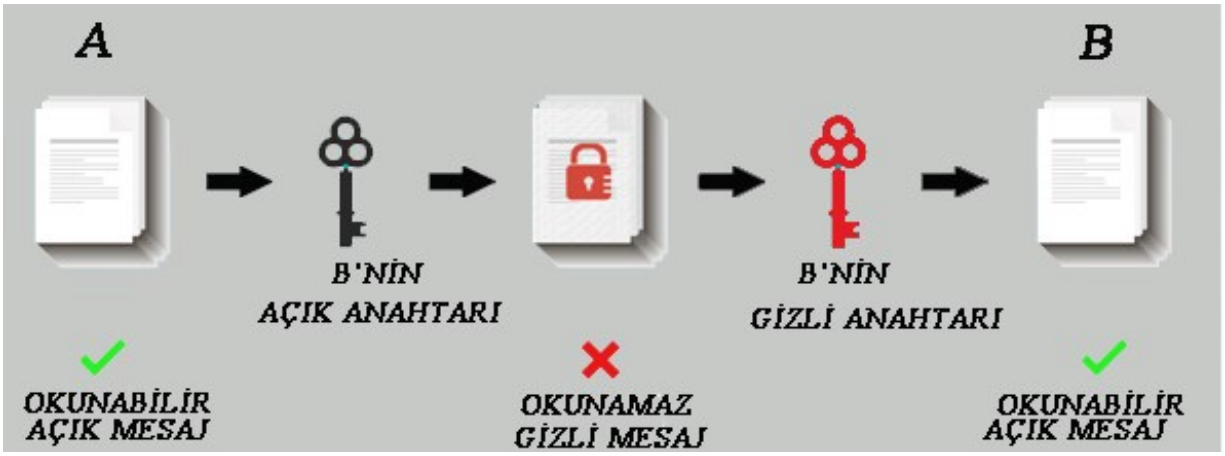
Simetrik şifrelemenin dezavantajları ise aşağıda sıralanmıştır [56]:

- Anahtarın saklanması zordur.
- n sayıda kullanıcı için n * (n-1) adet anahtarın saklanması gerekliliği vardır.
- Anahtarın güvenli bir şekilde dağıtımı problem teşkil etmektedir.
- Kimlik doğrulaması gerçekleştirmez. Bu sebepten dolayı anahtara sahip herhangi bir kişi tarafından veri şifrelenmiş olabilir. Göndericinin doğru kişi olduğunu teyit etmez.
- Gönderici ve alıcı arasındaki trafiği dinleyen kişi tarafından verinin değiştirilme durumu söz konusu olabilir. Verinin bütünlüğünü garanti etmez.

5.3.2. Asimetrik Şifreleme

Simetrik algoritmayı kullanarak şifreleme yapan çok kullanıcılı bir sistemde bütün kullanıcıların aynı şifreyi kullanması veri güvenliğini tehlikeye atarken; her bir kullanıcı için farklı anahtarların üretilmesi sorunlara yol açabilmektedir [57]. Çok kullanıcı içeren bir sistemde, kullanıcıların bir birileriyle güvenli haberleşme gerçekleştirebilecek kullanıcı sayısının fazlalığı göz önünde bulundurulduğunda; simetrik şifreleme her bir gönderici ve alıcı çifti arasında bir anahtarın oluşturulması ve bunların güvenli bir şekilde paylaşılması gibi çeşitli güvenlik sorunlarına neden olmaktaydı. Asimetrik şifreleme bu tür sorunların çözümü için geliştirilmiştir.

Asimetrik şifrelemede iki adet anahtar kullanılmaktadır. Her bir kullanıcının kendine ait bir genel anahtarı, bir de özel anahtarı bulunmaktadır. Genel anahtar şifreleme işleminde kullanılıp; özel anahtar ise alınan şifreli mesajın çözülme sürecinde kullanılır. Gönderici alıcı tarafa bir mesaj göndermek istediğinde; mesajını alıcının genel anahtarı ile şifreler ve alıcıya gönderir. Alıcı aldığı şifreli mesajı kendi özel anahtarı ile çözer. Asimetrik şifreleme yönteminin basamakları Şekil 5.3'te gösterilmiştir.



Şekil 5.3. Asimetrik şifreleme basamakları [64]

Asimetrik şifrelemede gönderici, alıcının genel anahtarıyla şifrelediği verinin sadece alıcının özel anahtarıyla deşifre edebileceğini bilir. Ayrıca göndericinin kendi özel anahtarıyla şifrelediği bir veri alıcı tarafında, göndericinin genel anahtarı ile deşifre edilerek bu verinin gerçekten göndericiden gelip gelmediğinden emin olabilir.

Asimetrik algoritmaların avantajlı tarafları şu şekildedir [57]:

- Veri bütünlüğü, gizliliği ve kimlik doğrulama işlemlerini güvenli bir biçimde gerçekleştirirler.
- Anahtar kullanıcı tarafından belirlenebilir.

Asimetrik algoritmaların dezavantajlı tarafları şu şekildedir [57]:

- Algoritmaların yavaş çalışıyor olması.
- Anahtar uzunluklarının problemlere neden olabilmesi.

RSA, DSA ve ECDSA algoritmaları asimetrik şifreleme yöntemlerine örnek olarak gösterilebilir.

5.3.3. Hibrit Şifreleme

Hibrit şifreleme de olarak adlandırılan ve bazı uygulamalarda simetrik ve asimetrik şifrelemelerin birbirlerine karşı üstünlerini kullanmak için bu iki algoritma birlikte kullanılmıştır. Bu üstünlükler güvenlik ve hızlılıktır. Bu yöntemler genellikle gönderici tarafından üretilen simetrik şifreleme anahtarının alıcının genel anahtarı ile şifrelenerek alıcıya gönderilmesi, alıcının gelen mesajı kendi özel anahtarı ile deşifreleyerek simetrik şifreleme anahtarına ulaşması esasına dayanır. Anahtar güvenli bir şekilde alıcıya gönderildikten sonra işlemlerin daha hızlı gerçekleştirilmesi için simetrik şifreleme kullanılır.

5.4. KMAA'larda Şifreleme

KMAA'larda kullanılan algılayıcıların kısıtlı bir enerji barındıran dahili bir bataryadan beslendiği daha önceki bölümlerde ifade edilmişti. Bu tez çalışmasında uygulamanın daha uzun ömürlü olabilmesi için enerji verimliliği ve işlem süresi açısından iyi performans gösteren "özel veya - XOR" şifrelemeden yararlanılmıştır. XOR şifreleme yoğun hesaplamalar gerektirmediği, mantıksal düzeyde işlem gerçekleştirdiği ve bilgisayar sistemlerinde uygulanması kolay olduğu için tercih edilmiştir.

5.4.1. XOR Şifreleme

Simetrik bir şifreleme algoritması olan XOR şifrelemenin doğal yapısı ile belirli uzunluktaki verinin şifrelenmesi hızlı ve kolay bir şekilde gerçekleştirilebilir. XOR şifreleme kaba kuvvet saldırılarıyla kırılması zordur [58]. XOR işlemi $\oplus$ sembolü ile gösterilir. XOR işleminin giriş ve çıkışları Tablo 5.1'de verilmiştir.

Tablo 5.1. XOR giriş ve çıkışları

Giriş		Çıkış
A	B	C
0	0	0
1	0	1
0	1	1
1	1	0

Tablo 5.1'den görüleceği üzere XOR kapısının girişlerinin aynı değerlere sahip olması durumunda çıkış sıfır; diğer durumlarda ise bir olmaktadır.

XOR İşleminin Özellikleri

XOR işleminin değişme özelliği vardır. Bu durum Denklem 5.1'de gösterildiği gibi ifade edilebilir.

$$A \oplus B = B \oplus A \tag{5.1}$$

Bir değerin sıfır ile XOR işlemine tabi tutulması sonucunda sonuç değerin kendisi olacaktır. Bu durum Denklem 5.2'de gösterildiği gibi ifade edilebilir.

$$A \oplus 0 = A \tag{5.2}$$

Bir değerin kendisi ile XOR işlemine tabi tutulması sonucunda çıkış sıfır olacaktır. Bu durum Denklem 5.3'te gösterildiği gibi ifade edilebilir.

$$A \oplus A = 0 \tag{5.3}$$

İki değerin XOR işlemine tabi tutulması durumunda oluşan sonucun; XOR işleminin giriş değerlerinden biriyle tekrar XOR işlemine tabi tutulması durumunda elde edilen değer, diğer giriş değeridir. Bu durum Denklem 5.4'te gösterildiği gibi ifade edilebilir.

$$C \oplus A = B$$

$$C \oplus B = A \tag{5.4}$$

Denklem 5.4 XOR işleminin daha önce anlatılan özelliklerinden faydalanılarak elde edilebilir.

Denklem 5.4 'ten de görüleceği üzere şifrelenmek istenen bir A verisinin B anahtarı ile XOR işlemine tabi tutulması sonucunda oluşan sonuç C, girişten farklı olup; şifreli mesaj olarak nitelenebilir. Şifreli mesajın alıcı tarafta, alıcı ve göndericinin bildiği anahtar ile XOR işlemine tabi tutulması sonucunda asıl mesaja ulaşılabilir. XOR şifrelemede, şifreleme ve deşifreleme işlemlerinde kullanılan program veya fonksiyon aynıdır.

XOR şifrelemede kullanılan anahtarın boyutu şifrelemenin güvenlik düzeyini etkiler. Sabit boyutlu bir anahtar kullanılması durumunda, mesajı kırmak isteyen herhangi bir kişi frekans analizi ile asıl mesaja ulaşabilir [59]. Mesajla eşit uzunluktaki bir anahtarla ile verinin XOR şifreleme ile şifrelenmesi, deneme yanılma yöntemi ile verinin elde edilme ihtimalini azaltır. Bu durumda verinin güvenli iletimi kadar zor olan anahtarın iletilmesi problemini doğurur. XOR şifrelemede kullanılan anahtarın her defasında değişmesi şifrenin kırılmasını oldukça zorlaştırır.

Aynı anahtar şifreleme yapılması durumunda ortamı dinleyen ve şifreli mesajları ele geçiren bir kişi bu mesajları XOR işleminin matematiksel özelliklerinden yararlanarak deşifre edebilir [60]. Denklem 5.5'te gösterildiği gibi A ve D mesajlarının B anahtarı ile şifrelendiğini varsayalım.

$$A \oplus B = C$$

$$D \oplus B = E \tag{5.5}$$

Ortamı dinleyen kişinin C ve E mesajlarını okuduğunu varsayalım. Bu mesajları XOR işlemine tabi tuttuğunda Denklem 5.6'da gösterildiği gibi bir durum elde edilir.

$$\mathrm{C} \oplus \mathrm{E} = \mathrm{A} \oplus \mathrm{B} \oplus \mathrm{D} \oplus \mathrm{B} \tag{5.6}$$

XOR işleminin değişme özelliği kullanılır ise; bu durumda işlem Denklem 5.7'de gösterildiği gibi düzenlenebilir.

$$\mathrm{C} \oplus \mathrm{E} = \mathrm{B} \oplus \mathrm{B} \oplus \mathrm{A} \oplus \mathrm{D} \tag{5.7}$$

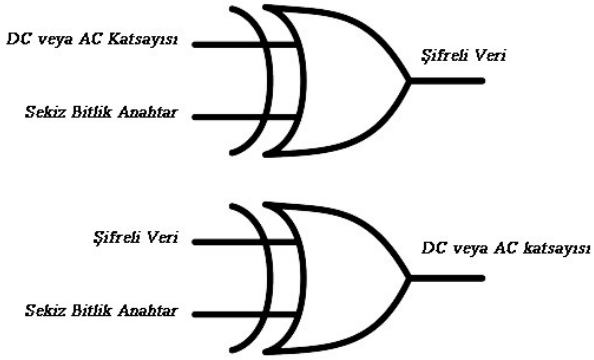
XOR işleminde $B \oplus B=0$ ve bir girişin sıfır ile XOR işlemine tabi olması, o değerin kendisini verdiği kurallar dikkate alındığında Denklem 5.7; Denklem 5.8'de gösterildiği gibi olur.

$$C \oplus E= A \oplus D \tag{5.8}$$

Denklem 5.8'den görüleceği üzere aynı anahtarın kullanılması şifrelenmemiş veriye ulaşma noktasında ortamı dinleyen bir kişiye büyük kolaylık sağlamaktadır [60]. Dolayısıyla şifreleme için tek kullanımlık anahtarların üretilmesi güvenlik açısından uygun olacaktır.

XOR işlemi ile şifreleme mantıksal düzeyde işlem gerektirdiği, karmaşık bir işlem olmadığı için oldukça hızlıdır. Bu tez çalışmasında KMAA'ların yaşam döngüsünün daha uzun olması ve dolayısıyla şifreleme esnasında daha az enerji tüketildiği için XOR işlemi ile şifreleme tercih edilmiştir.

Bu çalışmada, herhangi bir fotoğrafın şifrelenmesi, fotoğrafın herbir 8x8 boyutundaki blokları için AKD ve niceleme süreçlerinden sonra elde edilen DC katsayıları veya DC katsayılarıyla beraber bazı AC katsayılarının, rastgele oluşturulacak 8 bitlik anahtar değeri ile XOR'lanması ile gerçekleştirilmiştir. Bu yapıyla ilgili şifreleme ve deşifreleme şemaları Şekil.5.4'te verilmiştir.



Şekil 5.4. Şifreleme ve deşifreleme şeması

6. MATERYAL VE METOT

Bu tez çalışmasında KMAA'larda kullanılmak üzere, JPEG sıkıştırma tekniğine dayanan literatürde LEICA ismiyle anılan düşük enerjili resim sıkıştırma algoritması detaylıca incelenip; ön plandaki değişim bölgelerinde farklı sıkıştırma sınıfları oluşturularak, sıkıştırma için yapılan hesaplama sayısında ve sıkıştırma oranında LEICA yöntemine göre iyileştirmeler yapılmıştır. Buna karşılık resim kalitesi açısından önemli bir değişiklik olmadığı gözlenmiştir. Bu iyileştirilen yönteme Değişken-LEICA (D-LEICA) adı verilmiştir. Ayrıca havada korumasız bir şekilde iletilen verinin yetkisiz kişiler tarafından ele geçirilmesini engellemek amacı ile basit bir şifreleme işlemi gerçekleştirilmiştir. Bu şifreleme işleminin temelinde; sıkıştırılacak resmin her bir 8x8 boyutundaki bloklarında bulunan DC veya DC ve AC katsayılarının farklı sayıda anahtarlar ile XOR işlemine tabi tutulması vardır.. Kullanım amacına göre kullanıcının farklı düzeylerde şifreleme işlemi gerçekleştirebileceği gösterilmiştir.

Alt bölümlerde sırasıyla; önerilen D-LEICA sıkıştırma ve XOR şifreleme uygulamalarının MATLAB ortamında gerçekleştirilmesi ve testleri açıklanacaktır.

6.1. D-LEICA

Bölüm 4'te açıklaması yapılan LEICA algoritması ön plan resminde, sadece ρ_{maks} ve ρ_{min} iki farklı sınıf kullanarak sıkıştırma işlemi gerçekleştirmektedir. Yani algoritma eğer değişim tespit ederse; minimum sıkıştırma, eğer önemli derecede bir değişim tespit etmez ise maksimum sıkıştırma yapmaktadır. LEICA algoritması değişimin olduğu alanlarda bir sınıflandırma yapmamaktadır. Bu durumda sıkıştırılmış resmin boyutu çok fazla düşürülmemiş olur.

Oysaki ön plan resminde farklı oranda değişimin olduğu alanlar mevcuttur. Dolayısıyla ön plan resminde az değişim gösteren bölgelerin daha fazla sıkıştırılması veya tersi olması durumunda ilgili alanların daha az sıkıştırılması daha verimli olabilir. Bunun için D-LEICA algoritması LEICA algoritmasına göre sadece belirli ρ_{maks} ve ρ_{min} gibi iki değer kullanmak yerine; ρ_{maks} ve ρ_{min} değerleri arasında değişim oranına farklı seviyede değerler alabilen ρ değerlerini (minimum bir ve maksimum sekiz değeri) kullanır.

Bu işlemi daha açık ifade etmek gerekirse; LEICA algoritması blokları iki sınıfa ayırıp; buna göre sıkıştırma aşamalarını gerçekleştirmesine karşın; D-LEICA uygulamaya bağlı olarak ikiden fazla sınıf (kullanıcı tercihine veya uygulamaya göre 8'e kadar artabilir) kullanabilme özelliğine sahiptir. LEICA algoritması önemli olarak sınıflandırdığı bir blok için bir sıkıştırma modeli sunmasına karşılık; D-LEICA algoritması birden fazla sınıflandırma modeli sunabilir. Bu durum bloklar arasında bir önem derecesi oluşturmaya; en önemli bloğun minimum sıkıştırılmasına, derece olarak daha az öneme sahip blokların ise daha fazla sıkıştırılmasına olanak sağlar.

6.1.1. D-LEICA Algoritmasının Metadolojisi

D-LEICA algoritmasının da LEICA algoritmasında olduğu gibi bir eşik değeri (D_0) bulunmaktadır. Bu eşik değeri kullanıcı tarafından veya uygulamaya bağlı olarak belirlenen bir değerdir.

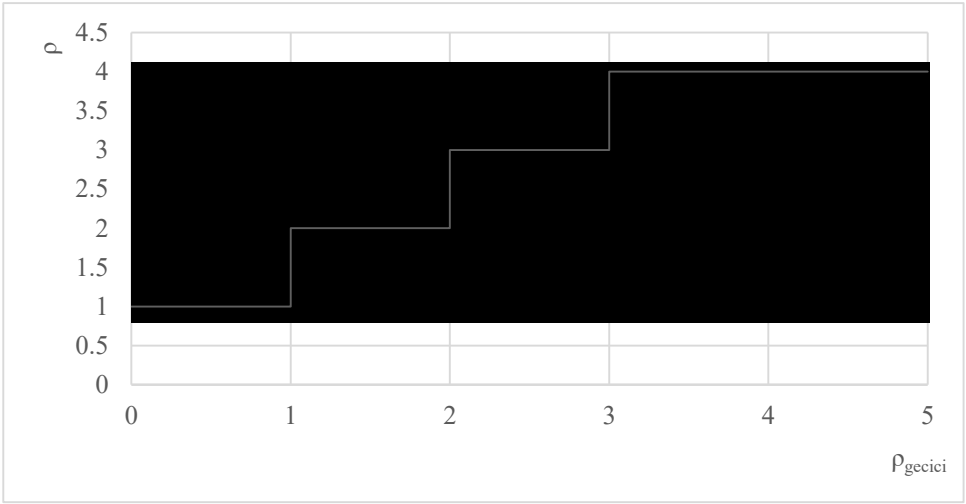
Algoritma ilk aşamada LEICA algoritmasının da kullanmış olduğu Denklem 4.2'deki hesaplamayı gerçekleştirir ve $D_{m,n}$ değerini elde eder. Daha sonra Denklem 6.1'de verilen hesaplamayı gerçekleştirir ve ρ_{gecici} değerini elde eder. Hesaplama sonucunda oluşan bu değer, bir blok için AKD sürecinde hangi ρ değerinin kullanacağını gösterir.

$$\rho_{gecici} = \frac{D_{\mathrm{m,n}}}{D_0} \tag{6.1}$$

Yöntem Denklem 6.1'de elde edilen ρ_{gecici} değeri için aşağıdaki kontrolleri yapar:

- Eğer elde edilen ρ_{gecici} değeri, ρ_{maks} değerinden büyük ise; o zaman sıkıştırma işleminde kullanılacak olan ρ_{gecici} değeri ρ_{maks} değerine eşitlenir.
- Eğer elde edilen ρ_{gecici} değeri ρ_{min} değerinden küçük oluyor ise; o zaman sıkıştırma işleminde kullanılacak olan ρ_{gecici} değeri ρ_{min} değerine eşitlenir.
- Eğer elde edilen ρ_{gecici} bir tam sayı değil ise; bu değer yukarı yuvarlama ile en yakın üst tam sayıya yuvarlanır.

ρ_{min} =1 ve ρ_{maks} =4 değerleri için D-LEICA algoritmasının bir resmin her bir bloku için hesaplanan ρ_{gecici} değerine göre oluşacak ρ'nun alabileceği değerler grafiği Şekil 6.1'de gösterildiği gibi olur.

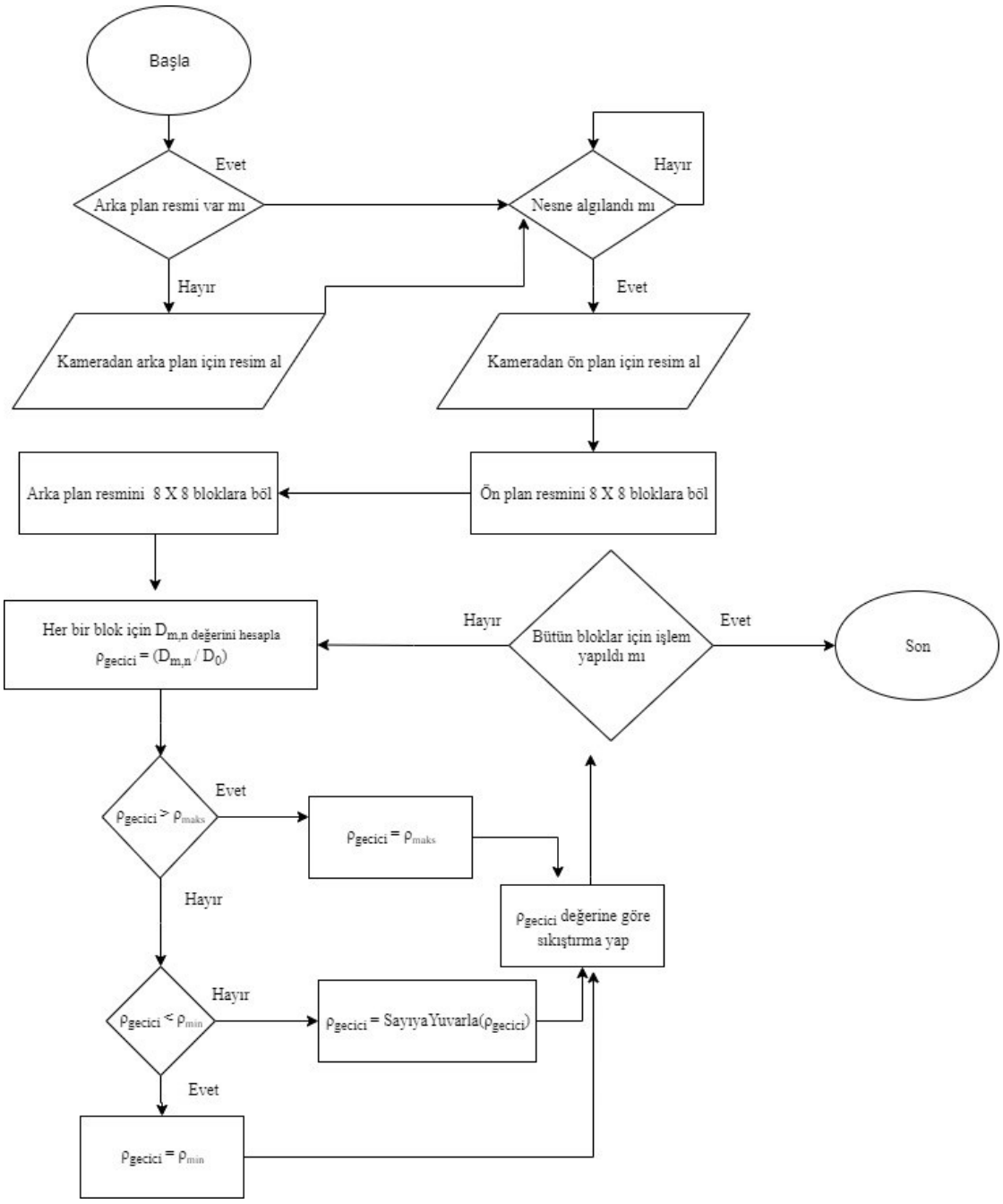


Şekil 6.1. ρ değerinin değişim grafiği

ρ_{gecici} değeri elde edildikten sonra, bu değere göre AKD süreci işletilir. Sonraki süreçler JPEG ve LEICA algoritmasında kullanılan süreçlerin aynısıdır.

6.1.2. D-LEICA Algoritmasının Akış Şeması

Bu tez çalışmasında önerilen D-LEICA algoritması Şekil 6.2'de gösterilen akış şeması esas alınarak MATLAB ortamında gerçekleştirilmiştir.



Şekil 6.2. D-LEICA algoritmasının akış şeması

Şekil 6.2'den görüleceği üzere D-LEICA, LEICA'dan farklı olarak bir ara değer olan ρ_{gecici} değerini hesapladıktan sonra, belirli kriterlere göre bir karar süreci işletmektedir.

6.2. Fotoğraf Şifreleme

Bu bölümde sıkıştırılan fotoğrafın güvenli bir şekilde ilgili yere iletilmesini sağlamak için XOR işlemi kullanılarak fotoğrafın şifrelenmesi sağlanmıştır. XOR işlemi ile bir verinin şifrelenebilmesi için öncelikle anahtar üretimine ihtiyaç vardır.

6.2.1. Anahtar Üretimi

XOR şifrelemede seçilen anahtarın, güvenlik düzeyini önemli ölçüde etkilediği Bölüm 5.4'te ifade edilmişti. Eğer yüksek güvenlikli bir uygulama için şifreleme yapılacaksa; kullanılacak olan anahtarların sayısı ve sırası, ortam dinlemesi yapan bir saldırganın tahmin edemeyeceği bir karmaşıklıkta gerçekleştirilmelidir. Aksi takdirde kaba kuvvet saldırılarıyla açık veri elde edilebilir. Anahtar üretimi ve paylaşımı için çeşitli uygulamalar olmasına karşın; bu çalışmada gerçekleştirilen uygulamada alıcı ve gönderici tarafın aynı algoritmayla anahtarlar üretebileceği bir modelin var olduğu kabul edilmiştir. Böylece önceden bilinen bir algoritmaya göre anahtarlar üretilmiş olup; üretilen bu anahtarların alıcı ve gönderici arasında paylaşılmasına ihtiyaç duyulmamıştır.

6.2.2. Şifreleme Süreci

Ayrık kosinüs dönüşümünün 8 X 8 boyutundaki bloklar üzerinde uygulandığı ve dönüşüm sonucunda oluşan matrisin birinci satır ve sütunda bulunan değerin DC katsayısı olarak ifade edilir. DC katsayısı bir bloğun ortalama değeri olup; o blok hakkında en önemli bilgiyi içermektedir.

Bu çalışmada gerçekleştirilen şifreleme uygulamasında; görüntü sıkıştırma aşamaları olan AKD, niceleme ve zikzak taramadan geçen bir 8 X 8 boyutundaki blok matrisi, XOR işlemi içeren bir şifreleme metoduyla şifrelenerek RLE ve Huffman kodlaması ile sıkıştırılmaktadır. Şifreleme için her bir blokta bulunan DC katsayısının rastgele üretilen sekiz bitlik anahtar veya anahtar grubuyla şifrelenmesi amaçlanmıştır. Rastgele üretilen anahtarların, aynı algoritmaya göre her iki tarafta üretildiği varsayılmıştır.

Uygulamaya bağlı olarak bazı AC katsayılarının şifrelenmesi güvenlik düzeyini arttıracağı gibi bu durum oluşacak şifreli görüntünün boyutunu arttıracaktır. KMAA'larda veri iletiminde kullanılan enerjinin yoğunluğu dikkate alındığında, görüntü boyutunun çok fazla artmadan şifreleme işleminin gerçekleştirilmesi sistem ömrünün uzun olması açısından önemli bir metrik olarak değerlendirilmektedir.

Gerçekleştirilen şifreleme uygulamasının akış şeması Şekil 6.3'te verilmiştir.

Şekil 6.3. Şifreleme yönteminin akış şeması

7. BULGULAR VE TARTIŞMA

Bu bölümde örnek olarak alınan 512 X 512 boyutunda standart Lena ve bilgisayarın kamerasından çekilen 640 X 480 boyutunda gri seviye fotoğrafların sırasıyla Bölüm 4, 6 ve 3'te açıklaması yapılan LEICA, D-LEICA, JPEG modellerine göre MATLAB ortamında sıkıştırılması testleri gerçekleştirilerek sonuçlar verilmiştir. Bütün testlerde kullanılan ρ_{min} değeri bir olup; ρ_{maks} değeri (iki ile sekiz arasında) değiştirilmiştir. Ayrıca fotoğraf şifrelenmesi için çeşitli testler gerçekleştirilmiştir.

7.1. Fotoğraf Sıkıştırma

Bu bölümde gerçekleştirilen testlerde LEICA, D-LEICA ve JPEG algoritmalarının sonuçları değerlendirilmiştir. Testlerde tüm algoritmalar için kullanılan niceleme matrisi standart Q_{50} matrisidir. Testlerde eşik değeri (D_0) ile ρ_{maks} ve ρ_{min} değeri LEICA algoritması ve D-LEICA algoritması için aynı olarak alınmıştır. İki algoritmanın benzer bir durumdaki işlem zamanı ve sıkıştırma oranı göz önünde bulundurulmuş ve algoritmaların karşılaştırılması yapılmıştır.

7.1.1. Sıkıştırma Oranına Göre Karşılaştırma

Bu bölümde gerçekleştirilen testlerde sırasıyla LEICA, D-LEICA ve JPEG yöntemlerine göre sıkıştırılan resimlerin boyutlarına ve görüntü kalitelerine göre bir karşılaştırma gerçekleştirilmiştir.

Şekil 7.1'de sonraki birkaç test işlemi için kullanılan arka plan ve ön plan fotoğrafları gösterilmiştir. Bu fotoğraflar esas alınarak sıkıştırma işlemleri gerçekleştirilmiştir. Arka plan olarak alınan fotoğraf, bir fotoğraf düzenleme programında standart Lena fotoğrafının ilgili kısmının çıkarılması sonucunda oluşturulmuştur.

Arka plan Ön plan

Şekil 7.1. Test için kullanılan fotoğraflar

Şekil 7.2'de sırasıyla LEICA VE D-LEICA algoritmalarının D_0=1 ve ρ_{maks}=4 değerlerine göre elde edilen sonuç verilmiştir.

Şekil 7.2. D_0=1 ve ρ_{maks}=4 değerleri için elde edilen sonuç

Şekil 7.2'den görüleceği üzere D-LEICA, LEICA algoritmasına göre %2 daha iyi bir sıkıştırma sağlamış olmasına karşın elde edilen görüntülerin görüntü kaliteleri neredeyse aynıdır. Yine Şekil 7.2'den en iyi görüntü kalitesine sahip olan ve görüntü boyutunun en fazla olduğu yöntemin JPEG olduğu görülmektedir.

Tablo 7.1'de D_0=1 ve buna bağlı olarak değişen ρ_{maks} değerlerine göre LEICA ve D-LEICA algoritmalarının sonuçlarının boyut ve PSNR değerleri verilmiştir. Elde edilen sonuçlardan görüleceği üzere ρ_{maks} değerinin artarken; algoritmaların oluşturduğu sonucun boyutu da artmaktadır.

Tablo 7.1. D_0=1 için elde edilen test sonuçları

	LEICA		**D-LEICA**	
ρ_{maks}	**BOYUT**	**PSNR**	**BOYUT**	**PSNR**
2	8547	26,55	8426	26,48
3	12694	28,31	12447	28,13
4	15943	29,35	15582	29,1
5	17565	29,82	17130	29,52
6	17915	29,96	17470	29,64
7	17943	29,97	17497	29,66
8	17949	29,98	17502	29,66

Şekil 7.3'te LEICA ve D-LEICA algoritmalarının kullandığı değerlerin D_0=3 ve ρ_{maks}=4 olması durumunda gerçekleştirilen test işlemine ait sonuçlar gösterilmiştir.

Şekil 7.3. D_0=3 ve ρ_{maks}=4 değerleri için elde edilen sonuç

Şekil 7.3'teki sonuçlar ile Şekil 7.2'deki sonuçlar kıyaslandığında; eşik değerinin artmış olması LEICA ve D-LEICA algoritmalarının oluşturduğu sıkıştırılmış veri boyutunu azaltmıştır. Ayrıca D-LEICA'nın LEICA algoritmasına göre yaklaşık % 3 daha iyi bir sıkıştırma oranı sağladığı görülmektedir. İki algoritmanın oluşturduğu sonuçların görüntü kalitesi arasında önemli bir farklılık bulunmamaktadır.

Tablo 7.2'de iki algoritmanın D_0=3 ve değişen ρ_{maks} değerlerine göre elde edilen sonuçları verilmiştir.

Tablo 7.2. D_0=3 için elde edilen test sonuçları

	LEICA		**D-LEICA**	
ρ_{maks}	**BOYUT**	**PSNR**	**BOYUT**	**PSNR**
2	8385	26,45	8352	26,42
3	12407	28,09	12229	27,95
4	15570	29,05	15162	28,81
5	17147	29,47	16598	29,17
6	17491	29,6	16920	29,28
7	17518	29,61	16945	29,29
8	17524	29,62	16950	29,29

Tablo 7.2 ve Tablo 7.1'deki sonuçlar kıyaslandığında; eşik değerinin artmış olması, oluşan sonucun boyutunu düşürmüştür. Bu durum sıkıştırma oranının arttığı anlamına gelir.

Şekil 7.4'te LEICA ve D-LEICA algoritmalarının kullandığı değerlerin D_0=5 ve ρ_{maks}=4 olması durumunda gerçekleştirilen test işlemine ait sonuçlar gösterilmiştir.

Şekil 7.4. D_0=5 ve ρ_{maks}=4 değerleri için elde edilen sonuç

Şekil 7.4'teki sonuç önceki sonuçlar ile kıyaslandığında; eşik değerinin bir miktar daha artmış olması; LEICA ve D-LEICA algoritmalarının oluşturduğu sonucun boyutunu azaltmıştır. Ayrıca eşik değerinin arttırılması sonucunda D-LEICA algoritmasının LEICA algoritmasına göre % 4 daha iyi bir sıkıştırma sağladığı görülmektedir.

Tablo 7.3'te iki algoritmanın D_0=5 ve değişen ρ_{maks} değerlerine göre elde edilen sonuçları verilmiştir.

Tablo 7.3. D_0=5 için elde edilen test sonuçları

	LEICA		**D-LEICA**	
ρ_{maks}	**BOYUT**	**PSNR**	**BOYUT**	**PSNR**
2	8299	26,36	8246	26,29
3	12249	27,91	11930	27,66
4	15344	28,81	14684	28,41
5	16892	29,2	15943	28,72
6	17230	29,31	16309	28,81
7	17258	29,33	16332	28,82
8	17263	29,33	16337	28,82

Şekil 7.5'te LEICA ve D-LEICA algoritmalarının kullandığı değerlerin D_0=10 ve ρ_{maks}=4 olması durumunda gerçekleştirilen test işlemine ait sonuçlar gösterilmiştir.

Şekil 7.5. D_0=10 ve ρ_{maks}=4 değerleri için elde edilen sonuç

Şekil 7.5'te elde edilen sonuçtan görüleceği üzere; D-LEICA LEICA'ya oranla yaklaşık %8 daha iyi bir sıkıştırma sağlamaktadır. Elde edilen görüntülerin kalitesi açısından D-LEICA ve LEICA algoritmalarının çıktıları arasında gözle görülür büyük bir farklılık görünmemektedir.

Tablo 7.4'te iki algoritmanın D_0=10 ve değişen ρ_{maks} değerlerine göre elde edilen sonuçları verilmiştir.

Tablo 7.4. D_0=10 için elde edilen test sonuçları

	LEICA		**D-LEICA**	
ρ_{maks}	**BOYUT**	**PSNR**	**BOYUT**	**PSNR**
2	8044	25,96	7993	25,92
3	11763	27,22	11252	27,02
4	14692	27,93	13528	27,57
5	16157	28,23	14507	27,77
6	16481	28,32	14754	27,83
7	16508	28,33	14470	27,83
8	16513	28,33	14774	27,83

Tablo 7.4'ten görüleceği üzere eşik miktarının bir miktar daha artmış olması algoritmaların oluşturduğu sonucun boyutunu düşürmüştür.

Şekil 7.6'da LEICA ve D-LEICA algoritmalarının kullandığı değerlerin D_0=20 ve ρ_{maks}=4 olması durumunda gerçekleştirilen test işlemine ait sonuçlar gösterilmiştir.

Şekil 7.6. D_0=20 ve ρ_{maks}=4 değerleri için elde edilen sonuç

Şekil 7.6'daki sonuçlardan görüleceği üzere eşik miktarının (D_0 = 20) bir miktar daha arttırılması durumunda; D-LEICA'nın LEICA'ya oranla yaklaşık %15 daha iyi bir sıkıştırma sağladığı görülmektedir. Görüntü kalitesi açısından D-LEICA ve LEICA algoritmaları değerlendirildiğinde; gözle görülür çok büyük bir farklılık görülmemektedir.

Tablo 7.5'te iki algoritmanın D_0=20 ve değişen ρ_{maks} değerlerine göre elde edilen sonuçları verilmiştir.

Tablo 7.5. D_0=20 için elde edilen test sonuçları

	LEICA		**D-LEICA**	
ρ_{maks}	**BOYUT**	**PSNR**	**BOYUT**	**PSNR**
2	7523	25,54	7470	25,49
3	10834	26,55	9928	26,23
4	13466	27,09	11292	26,51
5	14735	27,32	11909	26,58
6	15053	27,39	11970	26,58
7	15078	27,4	11970	26,58
8	15083	27,4	11970	26,58

Şekil 7.7'de LEICA ve D-LEICA algoritmalarının kullandığı değerlerin D_0=40 ve ρ_{maks}=4 olması durumunda gerçekleştirilen test işlemine ait sonuçlar gösterilmiştir.

Şekil 7.7. D_0=40 ve ρ_{maks}=4 değerleri için elde edilen sonuç

Şekil 7.7'den D-LEICA algoritmasının LEICA algoritmasına göre yaklaşık % 30 daha iyi bir sıkıştırma oranı sağladığı görülmektedir. Elde edilen PSNR değerleri incelendiğinde büyük bir farklılık söz konusu değildir.

Tablo 7.6'da iki algoritmanın D_0=40 ve değişen ρ_{maks} değerlerine göre elde edilen sonuçları verilmiştir.

Tablo 7.6. D_0=40 için elde edilen test sonuçları

	LEICA		D-LEICA	
ρ_{maks}	BOYUT	PSNR	BOYUT	PSNR
2	6505	24,92	6443	24,91
3	9027	25,57	7740	25,16
4	11050	25,92	7753	25,16
5	12044	26,06	7753	25,16
6	12318	26,11	7753	25,16
7	12342	26,11	7753	25,16
8	12347	26,11	7753	25,16

Tablo 7.6'dan görüleceği üzere eşik miktarının bir miktar daha arttırılmış olması oluşan sonuçların boyutunu bir miktar daha düşürmüştür.

Şekil 7.8'de LEICA ve D-LEICA algoritmalarının kullandığı değerlerin D_0=80 ve ρ_{maks}=4 olması durumunda gerçekleştirilen test işlemine ait sonuçlar gösterilmiştir.

.

Şekil 7.8. D_0=80 ve ρ_{maks}=4 değerleri için elde edilen sonuç

Şekil 7.8'den D-LEICA algoritmasının LEICA algoritmasına göre yaklaşık % 31 daha iyi bir sıkıştırma oranı sağladığı görülmektedir. Elde edilen PSNR değerleri incelendiğinde büyük bir farklılık söz konusu değildir.

Tablo 7.7'de iki algoritmanın D_0=80 ve değişen ρ_{maks} değerlerine göre elde edilen sonuçları verilmiştir.

Tablo 7.7. D_0=80 için elde edilen test sonuçları

	LEICA		D-LEICA	
ρ_{maks}	BOYUT	PSNR	BOYUT	PSNR
2	5385	24,2	5356	24,17
3	6715	24,42	5356	24,17
4	7806	24,53	5356	24,17
5	8339	24,58	5356	24,17
6	8492	24,59	5356	24,17
7	8498	24,59	5356	24,17
8	8502	24,59	5356	24,17

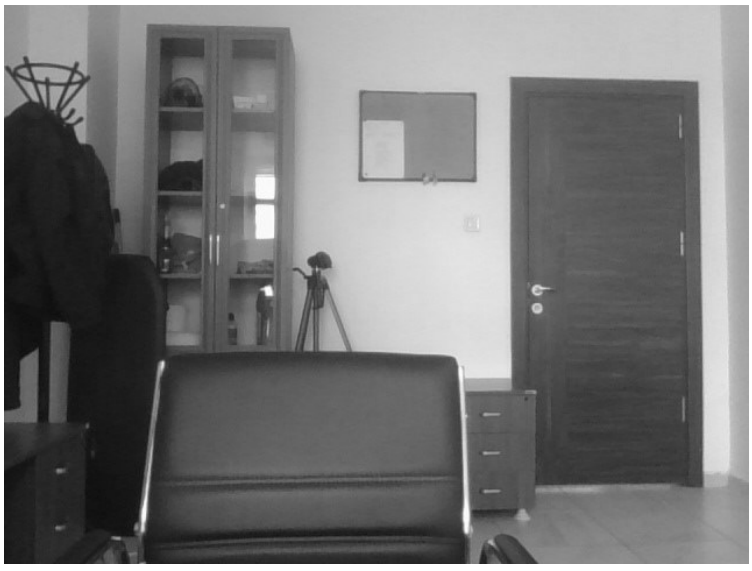
Tablo 7.7'den görüleceği üzere eşik miktarındaki artış oluşan görüntü boyutunun düşmesine neden olmaktadır. Eşik miktarının çok fazla arttırılması durumunda algoritmaların oluşturacağı sonuç Şekil 7.9'da verilmiştir.

Şekil 7.9. D_0=160 ve ρ_{maks}=4 değerleri için elde edilen sonuç

Şekil 7.9'daki sonuçtan görüleceği üzere çok büyük bir eşik değerinin kullanılması durumunda D-LEICA ve LEICA aynı sonuçları vermektedir. Eşik miktarı çok büyük seçildiği için, bütün veri blokları herhangi bir sınıflandırmaya tabi tutulamadığından; bütün veri blokları maksimum sıkıştırılmıştır.

Elde edilen sonuçlardan görüleceği üzere resmin boyutu üzerinde eşik miktarı ve ρ_{maks} değerinin önemli bir etkisi vardır. Bu sebepten dolayı kullanılacak bu değerlerin çok titiz bir şekilde uygulamaya bağlı olarak seçilmesi gerekmektedir.

Şekil 7.11 ve 7.12'de gerçekleştirilen testlerde kullanılan fotoğraflar bilgisayarın dahili kamerasından alınan (640 X 480 boyutundaki) görüntülerdir. Bu fotoğraflar sırasıyla LEICA, D-LEICA ve JPEG yöntemlerine göre sıkıştırılmış ve sıkıştırılmış verilerden görüntünün tekrar oluşturularak görüntülenmesi sağlanmıştır. Şekil 7.10'da gösterilen ekran görüntüsü bu testlerde kullanılan arka plan görüntüsünü göstermektedir.



Şekil 7.10. Arka plan fotoğrafı

Şekil 7.11 ve 7.12'de gerçekleştirilen testlerde kullanılan eşik değeri (D_0) bir metot tarafından otomatik bir şekilde hesaplanmıştır. Bu metot temelde arka plan fotoğrafı ve ön plan fotoğraflarının birinci (x=1, y=1 indeksindeki) blokların piksel değerlerinin toplamını ayrı ayrı hesaplamakta ve elde edilen bu değerlerin farkının ortalama değerini eşik değeri olarak almaktadır. Şekil 7.11'de ilk teste ilişkin sonuçlar verilmiştir.

Şekil 7.11. Kamera tarafından çekilen resmin sıkıştırılması

Şekil 7.11'de, D-LEICA'nın LEICA'ya oranla yaklaşık %17 daha iyi bir sıkıştırma sağladığı görülmektedir. D-LEICA'nın PSNR değeri LEICA algoritmasına göre daha düşük olmasına karşın; iki resim arasında gözle görülür büyük bir farklılık bulunmamaktadır.

Şekil 7.12'de başka bir test işlemine sonuç verilmiştir.

Şekil 7.12. ρ_{maks} değerininin arttırılması durumunda oluşan sonuç

Şekil 7.12'deki sonuçlardan görüleceği üzere D-LEICA algoritması LEICA algoritmasına göre yaklaşık %27 daha iyi bir sıkıştırma sağlamıştır.

7.1.2. İşlem Süresine Göre Karşılaştırma

Bu bölümde LEICA, D-LEICA ve JPEG yöntemlerine göre sıkıştırılan resimlerin işlem süresi yönünden bir karşılaştırma gerçekleştirilmiştir. Bu bölümde gerçekleştirilen testlerde standart Lena fotoğrafı kullanılmış olup bilgisayarda arka planda gerçekleştirilen işlemlerin hesaplama sonuçlarını etkilememesi için her bir algoritma yüz defa çalıştırılmıştır. Elde edilen değerler MATLAB programının sunmuş olduğu (Editor sekmesinde bulunan Run and Time butonunun çalıştırılmasıyla) bir araçla elde edilmiştir. Tablo 7.8'de üç algoritmanın da Lena fotoğrafını verilen parametrelere göre yüz defa işleme tabi tutmasında geçen toplam süre verilmiştir.

Tablo 7.8. Algoritmaların işlem sürelerinn değerlendirilmesi

ρ_{maks}	D_0	LEICA	D-LEICA	JPEG
3	**1**	**24,704**	**24,71**	**52,085**
3	3	24,143	24,102	50,306
3	**5**	**25,975**	**25,979**	**55,745**
3	10	24	23,41	50,287
3	20	30,796	29,93	64,79
3	40	23,595	23,172	52,997
3	**80**	**22,756**	**22,859**	**52,774**
4	1	35,545	34,686	69,386
4	3	33,27	33,043	63,83
4	**5**	**32,408**	**32,506**	**62,346**
4	10	34,014	33,055	66,229
4	20	32,104	30,473	64,667
4	40	30,561	29,195	65,155
4	80	34,285	33,281	74,244
5	1	37,078	35,975	64,715
5	3	29,638	29,251	52,584
5	5	34,736	34,093	62,007
5	10	34,304	32,242	61,931
5	20	32,965	30,498	61,727
5	40	31,935	29,029	63,387
5	80	28,842	27,084	61,904
6	1	38,569	37,906	61,049
6	3	38,884	37,876	63,813
6	5	38,124	36,494	63,231
6	10	34,856	32,058	59,797

Tablo 7.8'de 28 adet farklı test işlemine ait sonuçlar verilmiştir. Bu testlerin dört tanesinde (tabloda kalın yazılı satırlar) LEICA algoritması, D-LEICA algoritmasına göre daha az bir gerçekleştirme süresine sahip olmasına karşın; 24 adet testte D-LEICA en hızlı algoritma olarak öne çıkmıştır. Ortalama olarak D-LEICA, LEICA'ya oranla % 4 oranında daha hızlı işlem gerçekleştirmiştir. Tablo 7.8'de düşük ρ_{maks} değerinin kullanılması durumunda LEICA ve D-LEICA algoritmalarının gerçekleştirme süreleri bir birlerine çok yakın olmasına karşın; yüksek resim kalitesinin istendiği, dolayısıyla yüksek ρ_{maks} değerinin kullanıldığı uygulamalarda D-LEICA'nın işlem yönünden daha hızlı olacağı sonucuna varılmıştır.

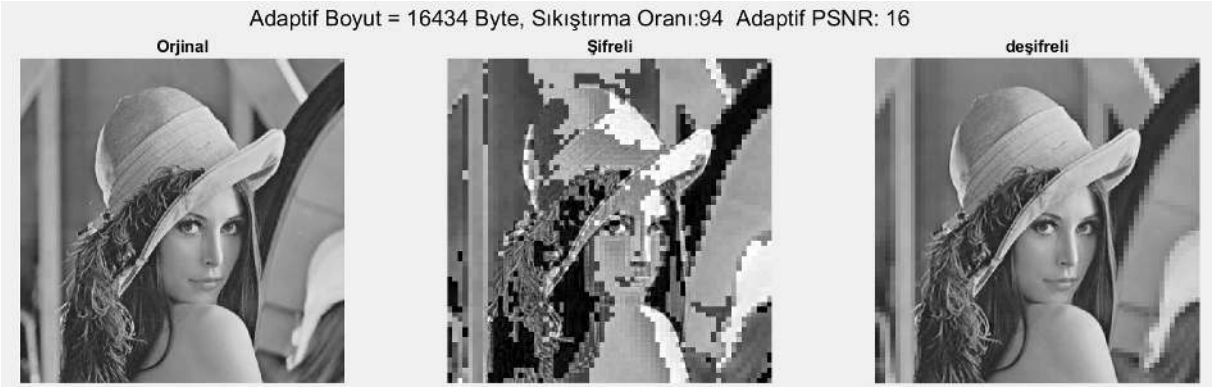
7.2. Şifreleme

Bu bölümde uygulanan testlerde, Bölüm 6'da açıklanan XOR şifreleme tekniği kullanılarak her bir blokun AKD sürecinden elde edilen DC katsayısının veya DC katsayısıyla beraber diğer AC katsayılarının şifrelenmesi sonuçları yorumlanacaktır.

Şekil 7.13'te standart Lena fotoğrafının 8 X 8'lik her bir bloğunda bulunan DC katsayısının bir adet rastgele oluşturulmuş sekiz bitlik anahtar değeriyle (150) XOR işlemine tabi tutulmasına ait örnek gösterilmiştir. Oluşan görüntü ve görüntünün boyutu, kullanılan anahtarın değerine göre farklılık gösterebilmektedir. Şekil 7.13'te ilk sütunda orijinal fotoğraf, ikinci sütunda orijinal fotoğrafın bir adet anahtar (150) ile şifrelenmiş hali ve son sütunda ise aynı anahtar ile şifreli fotoğrafın deşifre edilmiş hali gösterilmiştir.

Şekil 7.13. Bir adet DC katsayı anahtarı (150) ile Lena fotoğrafının şifrelenmesi

Şekil 7.14'te Lena fotoğrafının her bir bloğunun DC katsayısının sekiz bitlik bir adet anahtar değeri (30) ile şifrelenmesi sonucunda oluşan görüntü verilmiştir. Anahtar değerinin düşük bir değer alması durumunda oluşan şifreli görüntünün, orijinal görüntüye ait çok fazla detay barındırdığı görülmektedir. Bu şifreleme için istenen bir durum değildir.



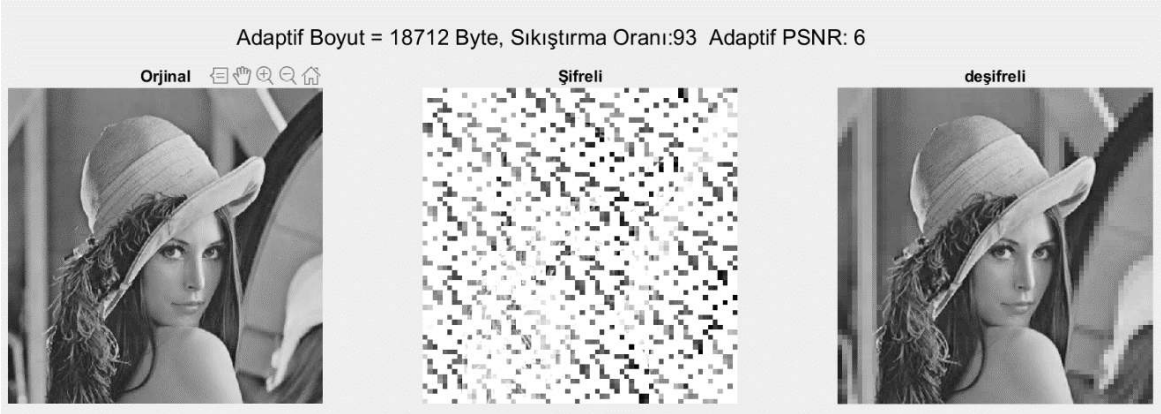
Şekil 7.14. Bir adet DC katsayı anahtarı (30) ile Lena fotoğrafının şifrelenmesi

Alıcı ve gönderici tarafta rastgele oluşturulmuş sekiz bitlik on adet anahtarın olduğunu ve bu anahtarlar ile fotoğrafın her bir bloğunda bulunan DC katsayılarının şifrelendiğini varsayalım. Bu duruma uygun olarak gerçekleştirilen bir uygulamaya ilişkin sonuç Şekil 7.15'te gösterilmiştir.

Şekil 7.15. On adet DC katsayı anahtar grubuyla görüntü şifreleme

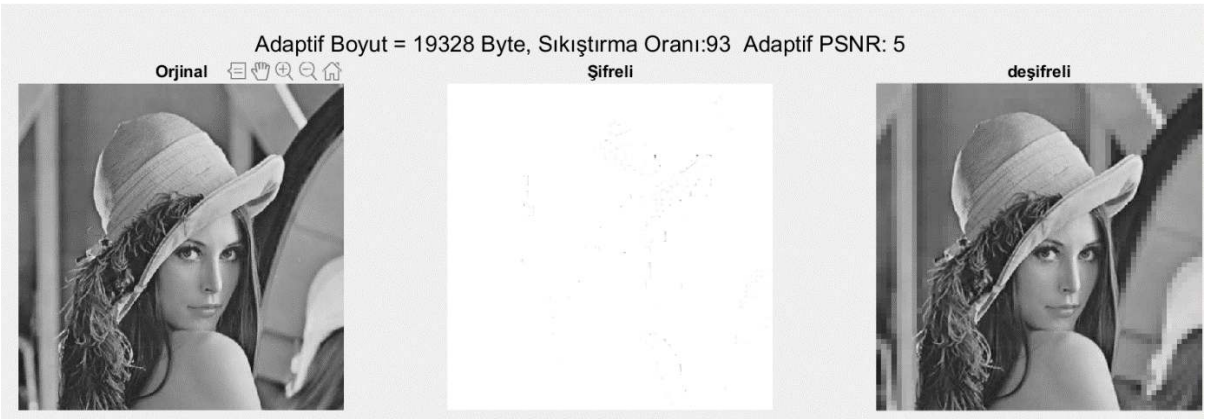
Şekil 7.15'te elde edilen görüntü kullanılan anahtarların değerine bağlı olarak değişkenlik gösterebilmektedir. Şifreli görüntünün orijinal görüntüye benzerlik göstermemesi için kullanılan sekiz bitlik anahtar değerleri rastgele olmak şartıyla üç haneli olarak alınmıştır.

Şekil 7.16'da rastgele bir şekilde üretilmiş sekiz bitlik yüz adetlik bir anahtar grubu kullanılarak fotoğrafın her bir bloğunun DC katsayısının şifrelenmesinin sonucu verilmiştir.



Şekil 7.16. Yüz adet DC katsayı anahtar grubuyla görüntü şifreleme

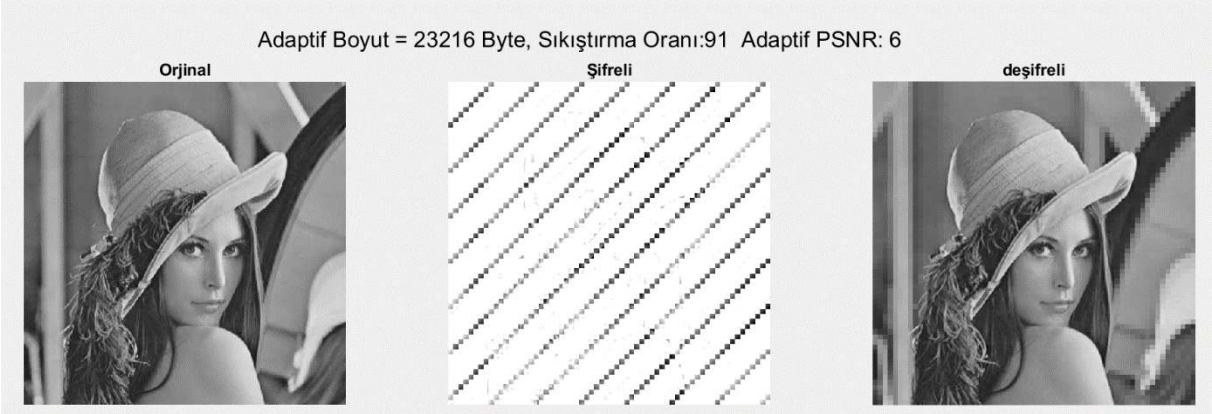
Şifreli fotoğrafın orijinal fotoğrafa olan benzerliğini azaltmak için DC katsayıları için yüksek değere sahip sekiz bitlik anahtar veya anahtar grubu oluşturulurken; AC katsayıları için oluşturulan anahtar veya anahtar grubunun değerleri düşük alınmıştır. Bu durum dikkate alınarak gerçekleştirilen şifreleme işleminde fotoğrafın her bir bloğunda bulunan DC katsayısı ve bu katsayının yanında bulunan iki adet AC katsayısını şifrelediğini varsayalım. Bu varsayım doğrultusunda rastgele seçilmiş olan birer adet sekiz bitlik DC katsayı anahtarı (227) ve AC katsayı anahtarı (3) için gerçekleştirilen test sonucu 7.18'de verilmiştir.



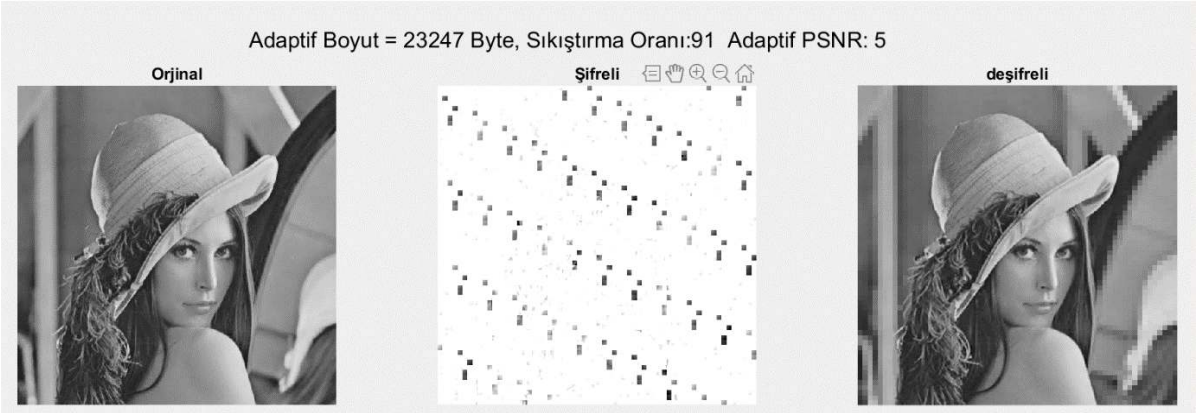
Şekil 7.17. Bir adet DC ve AC katsayı anahtarı ile görüntü şifreleme

Şekil 7.18 ile Şekil 7.13 ve 7.14'teki sonuçlar ile kıyaslandığında oluşan şifreli verinin boyutunun önemli bir oranda arttığı görülmektedir. Bu durum KMAA'larda uygulamaya bağlı olarak şifrelenecek olan katsayıların sayısının sistem ömrünü etkilediğini göstermektedir.

Şekil 7.19 ve 7.20'de sırasıyla sekiz bitlik on adet ve yüz adet rastgele oluşturulmuş DC ve AC anahtar gruplarıyla gerçekleştirilen şifreleme sonuçları verilmiştir. Elde edilen bu sonuçlar üretilen anahtarların değerine bağlı olarak değişkenlik gösterebilmektedir.



Şekil 7.18. On adet DC ve AC katsayı anahtar gruplarıyla görüntü şifreleme



Şekil 7.19. Yüz adet DC ve AC anahtar katsayı gruplarıyla görüntü şifreleme

Sadece DC katsayısının şifrelendiği örnekler ile DC katsayısı ve iki adet AC katsayısının şifrelendiği örneklerin sonuçları incelendiğinde, sadece DC katsayısının şifrelendiği örneklerde oluşan şifreli görüntünün boyutunun diğer örneklere göre önemli ölçüde düşük olduğu görülmektedir.

8. SONUÇLAR

Bu tez çalışmasında KMAA'larda kullanılmak üzere alternatif bir sıkıştırma algoritmasının gerçekleştirme ve test süreçleri gerçekleştirilmiştir. Kablosuz ortamda iletilen verinin istenmeyen kişiler tarafından ele geçirilmesini engellemek amacıyla bir şifreleme algoritması kullanılarak sıkıştırılan verinin güvenli şekilde iletilmesi amaçlanmıştır.

KMAA'ların sınırlı bir enerji kaynağı olan bir bataryadan enerji ihtiyacını karşılamaktadır. Ayrıca KMA'ların bulunduğu fiziksel çevreye ulaşmanın çoğu zaman zor olması nedeniyle bu cihazlara yeniden enerji verilmesi durumu da zordur. Bu sebepten dolayı sınırlı olan bu enerjinin verimli kullanılması problemi elzemdir.

Enerjinin verimli kullanılabilmesi için geleneksel sıkıştırma yöntemlerinin KMAA'lar için çeşitli uyarlamaları mevcuttur. JPEG sıkıştırma modelinin KMAA için bir uyarlaması olan LEICA algoritması bir resmi referans olarak aldığı bir resme göre (arka plan) bir sınıflandırma yapmak suretiyle sıkıştırma işlemi gerçekleştirmektedir.

Bu tez çalışmasında önerilen ve D-LEICA olarak isimlendirilen algoritma uygulamaya veya kullanıcı isteğine göre değişimin arka plan resmine göre fazla olduğu alanlarda bir sınıflandırma yapmak suretiyle resmin şifrelenmesini gerçekleştirmektedir.

MATLAB ortamında gerçekleştirilen testlerde LEICA, D-LEICA ve JPEG algoritmalarına göre sıkıştırılmış resimler veri boyutu ve resim kalitesine göre değerlendirilmiştir. LEICA ve D-LEICA algoritmalarının oluşturdukları resmin görüntü kalitesi JPEG algoritmasına oranla daha düşük olmasına karşın, veri boyutu olarak değerlendirildiğinde JPEG algoritmasına oranla çok daha iyi bir sıkıştırma gerçekleştirdikleri tespit edilmiştir.

LEICA ve D-LEICA algoritmalarının test edilmesinde kullanılan parametreler eşit tutulmuş olup; benzer parametrelere göre algoritmaların sonuçlarının ne olacağı tespit edilmeye çalışılmıştır. Test sonuçlarına göre kullanılan parametrelerin (ρ_{maks} ve D_0) sıkıştırma oranında doğrudan bir etkisi olduğu tespit edilmiştir.

Bilgisayar kamerasından değişken bir ortamda (ışık miktarının değiştiği) anlık olarak rastgele alınan resimlerin (640 X 480 boyutunda) LEICA, D-LEICA ve JPEG algoritmalarına göre sıkıştırılması durumunda elde edilen sonuca göre; D-LEICA'nın LEICA'ya oranla yaklaşık % 12 daha iyi bir sıkıştırma sağlamasına karşın; elde edilen resimlerin görüntü kalitesi açısından gözle görülür büyük bir farklılık bulunmamaktadır. LEICA algoritmasının elde ettiği görüntülerin ortalama PSNR değeri 33.33 olup D-LEICA'nın PSNR değeri 32.93'tür.

Standart 512 X 512 boyutundaki standart Lena resmi kullanılarak LECA, D-LEICA ve JPEG algoritmalarına göre sıkıştırma gerçekleştirilmesi durumunda (algoritmalar işletim sisteminden kaynaklı değişimleri minimize etmek amacıyla 100'er defa çalıştırılmıştır) işlem gerçekleştirme

süresine göre kıyaslandığında D-LEICA'nın yaklaşık LEICA'dan % 4 daha hızlı olduğu tespit edilmiştir.

Şifreleme için kullanılan anahtar sayısının fazla olduğu durumlarda elde edilen şifreli görüntünün; az sayıda anahtar kullanılarak gerçekleştirilen şifreli görüntüye oranla daha orijinal görüntüye daha az benzediği tespit edilmiştir. Kullanılan anahtarın boyutu sıkıştırma oranı üzerinde doğrudan bir etkisi vardır. Ayrıca bir blok verisinde şifrelenen katsayıların sayısının artması durumunda elde edilen şifreli görüntünün boyutu doğrudan artmaktadır. Bu sebepten dolayı kullanılacak uygulamaya göre kaç adet katsayının şifreleneceğinin belirlenmesi gerekmektedir.

ÖNERİLER

Bu tez çalışmasında yapılan testlerde eşik değeri (D_0) ve ρ_{maks} değerlerinin sıkıştırma oranı üzerinde doğrudan bir etkisi olduğu tespit edilmiştir. Sonraki çalışmalarda bu değerlerin seçimi noktasında; yapay zekâ uygulamalarından yararlanılması D-LEICA algoritmasının performansını doğrudan olumlu etkileyebileceği öngörülmektedir.

Ayrıca, basit bir şifreleme tekniği olan XOR şifreleme için anahtar değerinin seçimi şifreleme kalitesini etkilemektedir. Dolayısıyla, anahtar değerinin seçimini optimize edilebilirliğinin de araştırılması yerinde olur.

Bu çalışmada önerilen veri sıkıştırma ve şifreleme tekniklerinin, orta sınıf veya PDA sınıfı platformlarda kullanılabilirliği araştırmaya değer bir konu olarak öngörülmektedir.

KAYNAKLAR

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Akyildiz et al._2002_Wireless sensor networks a survey.pdf, Comput. Networks. 38 (2002) 393–422. www.elsevier.com/locate/comnet.

[2] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Comput. Networks. 52 (2008) 2292–2330. https://doi.org/10.1016/j.comnet.2008.04.002.

[3] Muhammad Zahid Khan, Ijaz Muhammad Khan, A Research Based Review of Wireless Sensor Networks, Ann. Comput. Sci. Ser. 9 (2011) 16.

[4] https://www.researchgate.net/figure/The-architecture-of-a-wireless-sensor-network-in-which-the-sensors-are-deployed-randomly_fig2_220719415, (2020). https://www.researchgate.net/figure/The-architecture-of-a-wireless-sensor-network-in-which-the-sensors-are-deployed-randomly_fig2_220719415.

[5] I.F. Akyildiz, T. Melodia, K.R. Chowdhury, Wireless multimedia sensor networks: applications and testbeds, Proc. IEEE. 96 (2008) 1588–1605. https://doi.org/10.1109/JPROC.2008.928756.

[6] N. Of, N. Of, O.R.K. At, W m s n : a s, Ieee Wirel. Commun. (2007) 32–39.

[7] https://www.researchgate.net/figure/Classification-of-image-compression-techniques_fig4_276481201, (15.01.2020).

[8] H. Zaineldin, M.A. Elhosseini, H.A. Ali, Image compression algorithms in wireless multimedia sensor networks: A survey, Ain Shams Eng. J. 6 (2015) 481–490. https://doi.org/10.1016/j.asej.2014.11.001.

[9] C.N. Taylor, S. Dey, Adaptive image compression for wireless multimedia communication, IEEE Int. Conf. Commun. 6 (2001) 1925–1929.

[10] A. Mammeri, A. Khoumsi, D. Ziou, B. Hadjou, Energy-aware JPEG for Visual Sensor networks, Mcseai. (2008). http://www.gel.usherbrooke.ca/khoumsi/Research/Public/MCSEAI08.pdf.

[11] A. Mammeri, A. Khoumsi, D. Ziou, B. Hadjou, Modeling and adapting JPEG to the energy requirements of VSN, Proc. - Int. Conf. Comput. Commun. Networks, ICCCN. (2008) 806–811. https://doi.org/10.1109/ICCCN.2008.ECP.151.

[12] A. Mammeri, A. Khoumsi, D. Ziou, B. Hadjou, Energy-efficient transmission scheme of JPEG images over visual sensor networks, Proc. - Conf. Local Comput. Networks, LCN. (2008) 639–647. https://doi.org/10.1109/LCN.2008.4664259.

[13] E. Sun, X. Shen, H. Chen, A low energy image compression and transmission in wireless multimedia sensor networks, Procedia Eng. 15 (2011) 3604–3610. https://doi.org/10.1016/j.proeng.2011.08.675.

[14] A. Senturk, R. Kara, An analysis of image compression techniques in wireless multimedia sensor networks, Teh. Vjesn. - Tech. Gaz. 23 (2016) 1863–1869. https://doi.org/10.17559/tv-20150313101530.

[15] D. Sora, Security Issues in Wireless Sensor Networks, Int. J. Online Eng. 6 (2010). https://doi.org/10.3991/ijoe.v6i4.1466.

[16] L.C. Han, N.M. Mahyuddin, An implementation of caesar cipher and XOR encryption technique in a secure wireless communication, 2014 2nd Int. Conf. Electron. Des. ICED 2014. (2011) 111–116. https://doi.org/10.1109/ICED.2014.7015781.

[17] R.M. Rad, A. Attar, R.E. Atani, A New Fast and Simple Image Encryption Algorithm Using Scan Patterns and XOR, Int. J. Signal Process. Image Process. Pattern Recognit. 6 (2013) 275–290. https://doi.org/10.14257/ijsip.2013.6.5.25.

[18] V. V Divya, S.K. Sudha, V.R. Resmy, Simple and secure Image Encryption, Int. J. Comput. Sci. Issues. 9 (2012) 286–289.

[19] L. Krikor, S. Baba, T. Arif, Z. Shaaban, Image encryption using DCT and stream cipher, Eur. J. Sci. Res. 32 (2009) 48–58.

[20] S.J. Xu, X.B. Chen, R. Zhang, Y.X. Yang, Y.C. Guo, An improved chaotic cryptosystem based on circular bit shift and XOR operations, Phys. Lett. Sect. A Gen. At. Solid State Phys. 376 (2012) 1003–1010. https://doi.org/10.1016/j.physleta.2012.01.040.

[21] I. Almalkawi, Wireless Multimedia Sensor Networks, Security and Key Management, (2013).

[22] I.T. Almalkawi, M.G. Zapata, J.N. al-Karaki, J. Morillo-Pozo, Wireless multimedia sensor networks: Current trends and future directions, Sensors. 10 (2010) 6662–6717. https://doi.org/10.3390/s100706662.

[23] H. Shen, G. Bai, Routing in wireless multimedia sensor networks: A survey and challenges ahead, J. Netw. Comput. Appl. 71 (2016) 30–49. https://doi.org/10.1016/j.jnca.2016.05.013.

[24] https://arkeofili.com/neandertaller-tarafindan-yapilmis-magara-resimleri-bulundu/, (15.01.2020).

[25] http://www.birkarefotograf.com/fotograf-nedir/, (15.01.2020).

[26] https://www.neatorama.com/2006/08/29/the-wonderful-world-of-early-photography/, (15.01.2020).

[27] https://www.wolframscience.com/reference/notes/1069b, (15.01.2020).

[28] https://en.wikipedia.org/wiki/Shannon–Fano_coding, (15.01.2020).

[29] https://en.wikipedia.org/wiki/LZ77_and_LZ78, (15.01.2020).

[30] S.M. Choudhary, A.S. Patel, S.J. Parmar, Study of LZ77 and LZ78 Data Compression Techniques, Certif. Int. J. Eng. Sci. Innov. Technol. 4 (2015) 45–49. http://www.ijesit.com/Volume 4/Issue 3/IJESIT201503_06.pdf.

[31] https://en.wikipedia.org/wiki/Lempel–Ziv–Welch, (15.01.2020).

[32] https://www.geeksforgeeks.org/lzw-lempel-ziv-welch-compression-technique/, (15.01.2020).

[33] http://techmeup.net/history-data-compression-infographic/, (15.01.2020).

[34] https://en.wikipedia.org/wiki/Zip_(file_format), (15.01.2020).

[35] https://hackernoon.com/why-do-we-need-jpeg-compression-and-how-its-technically-working-52a3a9ced55d, (15.01.2020).

[36] https://www.photokonnexion.com/definition-jpg-file-format/, (15.01.2020).

[37] https://link.springer.com/referenceworkentry/10.1007%2F0-387-30038-4_61, (15.01.2020).

[38] K.. G.P. CABEEN, İmage Compression and Discrete Cosine Transfom, (2020). https://www.math.cuhk.edu.hk/~lmlui/dct.pdf.

[39] V. Siraktamath, Impact of Quantization Matrix on the Performance of JPEG, 4 (2011) 107–118.

[40] W. M., M. Abdelrazek, Image Compression using DCT upon Various Quantization, Int. J. Comput. Appl. 137 (2016) 11–13. https://doi.org/10.5120/ijca2016908648.

[41] http://www.bretl.com/mpeghtml/zigzag.HTM, (15.01.2020).

[42] https://en.wikipedia.org/wiki/Run-length_encoding, (15.01.2020).

[43] https://www.prepressure.com/library/compression-algorithm/rle, (15.01.2020).

[44] http://compressions.sourceforge.net/Auxilary.html, (15.01.2020).

[45] A. Singh, An Enhanced Run Length Coding for JPEG Image Compression, Int. J. Comput. Appl. 72 (2013) 21–26.

[46] https://tr.wikipedia.org/wiki/Huffman_kodu, (15.01.2020).

[47] https://searchsecurity.techtarget.com/definition/cryptography, (15.01.2020).

[48] A Brief History of Cryptography, (15.01.2020). https://digitalgate.pw/crypto_history.htm?__cpo=aHR0cDovL3d3dy5jeXBoZXIuY29tLmF1.

[49] https://access.redhat.com/blogs/766093/posts/1976023, (15.01.2020).

[50] https://www.fizikist.com/muhtesem-bir-mucadele-enigma-makinesi-ve-alan-turing/, (15.01.2020).

[51] https://cs.stanford.edu/people/eroberts/courses/soco/projects/public-key-cryptography/history.html, (15.01.2020).

[52] http://bilgisayarkavramlari.sadievrenseker.com/2009/06/03/aes-ve-rijndeal-sifreleme/, (15.01.2020).

[53] https://digitalguardian.com/blog/what-data-encryption, (15.01.2020).

[54] https://pixelprivacy.com/resources/what-is-encryption/, (15.01.2020).

[55] https://kerteriz.net/modern-sifreleme-yontemleri-simetrik-asimetrik-sifreleme/, (15.01.2020).

[56] https://www.siberportal.org/blue-team/cryptography/basics-of-symmetric-encryption-and-asymmetric-encryption/, (15.01.2020).

[57] http://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/şifreleme-yöntemleri, (15.01.2020).

[58] https://www.codingunit.com/exclusive-or-xor-encryption, (15.01.2020).

[59] http://www.cs.grinnell.edu/~weinman/courses/CSC161/2019S/homework/xor-cipher.shtml, (15.01.2020).

[60] https://www.mehmetince.net/crypto-101-1-merhaba-exclusive-or-xor/, (15.01.2020).

ÖZGEÇMİŞ

Cebrail BARUT

KİŞİSEL BİLGİLER

Doğum Yeri	**: Elazığ**
Doğum Yılı	**: 1989**
Uyruğu	**:** T.C.
Adres	**:** Sunay Mah. Erzurum yolu Caddesi No:11/7 Merkez Muş
E-posta	**:** cebrail.barut@gmail.com
Yabancı Diller	**:** İngilizce (Düzey: B)

EĞİTİM BİLGİLERİ

Lisans	**:** Fırat Üniversitesi, Mhendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 2010
Lise	**:** Elazığ Lisesi, Elazığ,2010

ARAŞTIRMA DENEYİMİ

- ✓ Kullanılabilen bilgisayar programla diller: JAVA,CSharp,PHP,ASP,ASP.NET,C++
- ✓ Kullabildiğim programlar: Visual Studio, Netbeans, Eclipse, Office Uygulamaları, Photoshop, Fireworks

İŞ DENEYİMİ

B. YIL – 2011-	Muş İl Özel İdaresi Bilgi İşlem Müdürlüğü, Bilgisayar Mühendisi