

50256 Adet
Sarı

**EĞİTİMLİ SINIFLANDIRMA YÖNTEMLERİNİN
KARŞILAŞTIRILMASI**

MEHMET HALUK SEÇİLMİŞ

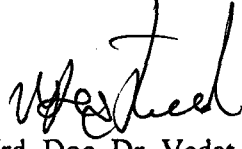
**Ç.Ü.
FEN BİLİMLERİ ENSTİTÜSÜ
FİZİK ANABİLİM DALI**

YÜKSEK LİSANS TEZ

**ADANA
ŞUBAT - 1996**

Çukurova Üniversitesi Fen Bilimleri Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından Fizik Anabilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.



Başkan : Yrd. Doç. Dr. Vedat PEŞTEMALCI



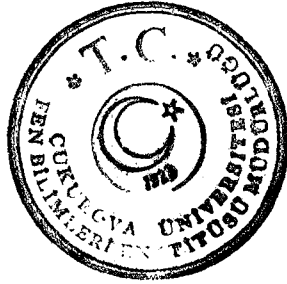
Üye : Prof. Dr. İlhami YEĞİNGİL



Üye : Yrd. Doç. Dr. Nuri EMRAHOĞLU

Kod No : 1096

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.



Prof. Dr. Ural DİNÇ
Enstitü Müdürü

Kirpiklerinize kadar sayabiliyorum...

Yeni gözlüğüyle annem.



Muray ve Eren 'e

İÇİNDEKİLER

S. NO

İÇİNDEKİLER	I
ÇİZELGE LİSTESİ.....	III
ŞEKİL LİSTESİ	IV
RESİM LİSTESİ	V
ÖZ.....	VI
ABSTRACT	VII
1. GİRİŞ.....	1
2. ÖNCEKİ ÇALIŞMALAR.....	3
3. MATERYAL ve METOD.....	7
3.1. Materyal.....	7
3.2. Metod	7
3.3. Sayısal Görüntü İşleme Konusunda Temel Prensipler	11
3.3.1. Görüntü.....	11
3.3.1.1. Sayısal Görüntü İşleme	11
3.3.1.2. Görüntünün Sayısallaştırılması	11
3.3.1.3. Sayısallaştırılan Görüntü Verilerinin Kaydedilmesi.....	13
3.3.2. Görüntü İşlemede Kullanılan İstatistiksel Özellikler	14
3.3.2.1. Görüntü Histogramı	14
3.3.2.2. Ortalama Değer ve Standart Sapma.....	15
3.3.3. Sınıflandırma	15
3.3.3.1. Eğitilmiş Sınıflandırma Yöntemleri	16
3.3.3.1.1. Maksimum Benzerlik Sınıflandırması.....	16
3.3.3.1.2. Minimum Mesafe Sınıflandırması.....	20
3.3.3.1.3. Kutu Sınıflandırması.....	21
3.3.3.1.4. Elipsoidal Sınıflandırma Yöntemi.....	22
3.3.3.1.5. En Yakın Komşu Yöntemi	22
3.3.3.1.6. Düşünülen Yeni Yöntem	23

4. ARAŞTIRMA BULGULARI.....	25
4.1. Sınıflama Öncesinde Yapılan Çalışmalar	25
4.1.1. Test Alanlarının ve Sınıfların Belirlenmesi	25
4.1.2. Sınıf Örneklerinin Belirlenmesi.....	26
4.1.3. Sınıf Örneklerine Ait Piksellerin İstatistiksel Değerlerinin Belirlenmesi.....	26
4.2. Sınıflandırma İşlemleri	33
4.2.1. Sınıflandırma Yöntemlerinin İşlem Hızlarının Karşılaştırılması	35
4.2.2. Yeni Algoritmanın (Üçgen Yöntemi) Başka Bir Veri Seti Üzerinde Denenmesi.....	35
5. TARTIŞMA ve SONUÇLAR	37
ÖZET	44
SUMMARY	45
KAYNAKLAR	46
TEŞEKKÜR.....	48
ÖZGEÇMİŞ	49
EK-1.....	50

ÇİZELGE LİSTESİ**S. NO**

Çizelge 1. Yer Gerçeğinin Belirlenmesinde Test Alanlarıyla Elde Edilen Sonuçlar ve Hata Oranları	26
Çizelge 2. Sınıfları Temsil Eden Örnek Alanlara Ait Piksellerin 3., 4., 5. Bantlardaki Ortalama ve Standart Sapma Değerleri	27
Çizelge 3. Sınıflandırma Sonuçları.....	33
Çizelge 4. Sınıflandırma Yöntemlerinin İşlem Süreleri	35
Çizelge 5. Eğitilmiş Sınıflandırma Sonucunda Elde Edilen Sınıfların Kapladıkları Alan Yüzdeleri	38
Çizelge 6. Üçgen Yöntemi İle Her Bir Örtü Tipi İçin Sınıflanamayan Piksel Yüzdeleri ve Sınıflanamayan Piksel Olasılıkları	38
Çizelge 7. Kutu Yöntemi İle Her Bir Örtü Tipi İçin Sınıflanamayan Piksel Yüzdeleri ve Sınıflanamayan Piksel Olasılıkları	39
Çizelge 8. Elipsoid Yöntemi İle Her Bir Örtü Tipi İçin Sınıflanamayan Piksel Yüzdeleri ve Sınıflanamayan Piksel Olasılıkları	39
Çizelge 9. Minimum Mesafe Yöntemi İle Her Bir Örtü Tipi İçin Sınıflanamayan Piksel Yüzdeleri ve Sınıflanamayan Piksel Olasılıkları	40
Çizelge 10. Maksimum Benzerlik Yöntemi İle Her Bir Örtü Tipi İçin Sınıflanamayan Piksel Yüzdeleri ve Sınıflanamayan Piksel Olasılıkları.....	40
Çizelge 11. En Yakın Komşu Yöntemi İle Her Bir Örtü Tipi İçin Sınıflanamayan Piksel Yüzdeleri ve Sınıflanamayan Piksel Olasılıkları.....	41
Çizelge 12. Eğitilmiş Sınıflandırma Yöntemlerinin Sınıflanamayan Toplam Piksel Yüzdesi ve Normalize Edilmiş Sınıflanamayan Toplam Piksel Olasılığına Göre Birbirleriyle İlişkileri	41
Çizelge 13. Eğitilmiş Sınıflandırma Yöntemlerinin Normalize Edilmiş Bilgisayar Zamanlarına Göre İşlem Hızları ve Toplam Performansları.....	42

ŞEKİL LİSTESİ**S. NO**

Şekil 1. Araştırmada İzlenen Akış Şeması.....	8
Şekil 2. Harita Üzerinde Yer Gerçeği Çalışması Yapılan Alanlar.....	10
Şekil 3. a) Sembolik Bir Görüntüye Ait Veri Seti, b) Veri Setindeki Piksel Değerleri ve Bu Piksellerin Sıklığı, c) Sembolik Görüntünün Histogramı	14
Şekil 4. Görüntünün 3 Banttaki Verileri ve Her Bantın Aynı Adresinden Sembolik Olarak Seçilen Sınıf Örnekleri.....	20
Şekil 5. Piksellerin Ortalama ve Standart Sapma Değerlerinin Kenarlarını Oluşturduğu Üçgenler	23
Şekil 6-a. Seçilen Sınıf Örneklerine Ait Piksellerin 3. Banttaki Dağılımı.....	28
Şekil 6-b. Seçilen Sınıf Örneklerine Ait Piksellerin 4. Banttaki Dağılımı.....	29
Şekil 6-c. Seçilen Sınıf Örneklerine Ait Piksellerin 5. Banttaki Dağılımı.....	30
Şekil 7. Seçilen Sınıf Örneklerinin a) 3. Bantta, b) 4. Bantta, c) 5. Bantta Ortalama Piksel Değerleri.....	31
Şekil 8. Seçilen Sınıf Örneklerine Ait Piksellerin Ortalamalarının Bantlara Göre Dağılımı	32

RESİM LİSTESİ**S. NO**

Resim 1. Çalışılan Alanın Landsat-TM Uydu Görüntüleri	9
Resim 2. Sınıflanmış Görüntüler	34
Resim 3. SPOT Uydusuna Ait Veri Setinin Ham ve Sınıflanmış Görüntüleri	36



ÖZ

Bu çalışmada, uzaktan algılama tekniği ile elde edilen verileri sınıflandırmak amacıyla geliştirilmiş eğitilmiş sınıflandırma yöntemleri (Maksimum Benzerlik, En Yakın Komşu, Minimum Mesafe, Kutu, Elipsoid ve Üçgen), doğruluk ve işlem hızları açısından birbirleriyle karşılaştırılmıştır. Bu yöntemler, Adana ilindeki tarımsal bir bölgeye ait LANDSAT-TM verileri üzerinde, hazırlanan bir görüntü işleme programıyla uygulanmıştır. Sınıflandırma işlemleri sonucunda oluşan sınıflara ait piksel sayıları ve bilgisayar süreleri değerlendirilmiş ve aradaki farklılıklar belirlenmiştir.



ABSTRACT

In this study, supervised classification methods (Maximum Likelihood, Nearest Neighbour, Minimum Distance, Box, Elipsoid and Triangle) to make classification for remote sensing data have been compared each other according to accuracy and computer processing time. These methods have been applicated on a LANDSAT-TM image of an agricultural area in Adana by home made programs about image processing. At the result of these classification processes, numbers of pixels of occurent classes and computer processing time have been determined and discussed differences among them.



GİRİŞ

Bilginin en çok önemi kazandığı çağımız, aynı zamanda bilgiye en hızlı bir şekilde erişilmesini de gerektirmektedir. Örneğin, üzerinde yaşadığımız yeryüzünden azami bir şekilde faydalanabilmek, tüm kitleyi kontrol edebilmek ve buna bağlı olarak geleceği planlama arzusu, bütün yeryüzünü tek bir çerçevede görebilme ihtiyacını ortaya çıkarmıştır.

Gelişen teknoloji sayesinde, yeryüzündeki nesneler üzerinden yansıyan elektromanyetik ışınımın, yeryüzünden yeterince yüksekliklerdeki yörüngelerde dolaşan uydular tarafından algılanıp, tayfsal analizinin değerlendirilmesi yoluyla bu nesneler hakkında bilgi toplanabilmektedir. Bu olaya Uzaktan Algılama (Remote Sensing) denir. Ancak, elektromanyetik ışınımın bazı dalga boyu bölgelerinde atmosferden geçerken soğurulması nedeniyle kayıpların oluşması, diğer bir deyişle beraberinde taşıdığı bilginin azalması söz konusudur. Dolayısıyla, uzaktan algılama olayında en önemli konulardan biri, yeryüzündeki farklı örtüleri ve nesneleri yine uzaktan algılama teknikleri ve araçları aracılığıyla elde edilen verileri kullanarak belirgin hale getirip, olabildiğince doğru olarak tanımlayabilmektir. Bilginin hızlı bir şekilde üretilmesi de ayrıca önem taşımaktadır.

Yapay uydular tarafından algılanıp yeryüzüne gönderilen veriler, bilgisayar sistemleri tarafından tekrar görüntüye dönüştürülüp bilgi üretilmektedir. Net bir görüntü ve doğru bilgi üretimi için, farklı görüntü zenginleştirme tekniklerini içeren görüntü işleme programları (GEMS, ERDAS gibi) ve görüntüyü dünya koordinatlarına uygun şekilde düzenleyen ve veri tabanının oluşmasında kullanılan Coğrafik Bilgi Sistemi (GIS-Geographic Information System) adı verilen programlar kullanılmaktadır. Bu programların genel işlevleri, oluşturulan görüntüler üzerinde coğrafik düzeltmeler yapmak, görüntüdeki yol, nehir gibi şekilsel özellikleri belirginleştirmek için filtreleme, tüm görüntüyü netleştirmek için, yoğunluk kesimi, doğrusal uzatma, ana bileşenler dönüşümü, ve histogram değişimi gibi teknikleri kullanarak renkleri ayrıştırma ve en önemlisi farklı bölgelerde aynı sınıftan olan yeryüzü örtülerinin belirlenmesi için görüntüyü değişik yöntemler kullanarak sınıflandırmak ve sonuçların istatistiksel

yorumunu yapmak şeklinde özetlenebilir.

Sınıflandırma, en genel anlamıyla, çok çeşitli nesnelerin rasgele bir araya gelerek oluşturduğu setin, aynı veya benzer özelliklere sahip olan elemanlarını ayrı ayrı gruplandırarak daha belirgin hale getirmektir. Uzaktan algılama olayında, bu elemanlar bir görüntüyü oluşturan piksellerdir. Bu işlemin en genel uygulaması, yeryüzünün belli bir bölgesindeki ekin alanları, su ve orman gibi değişik yeryüzü örtülerini sınıflamaktır. Sınıflandırma, aynı zamanda bir etiketleme problemi olarak da düşünülebilir. Bir bölgedeki her nesne bir sınıf adıyla (veya numarasıyla) etiketlenilirse, bölgenin detayları ortaya çıkar ve böylece ilgili karar işlemleri kolaylaşır.

Görüntüleri sınıflandırmanın temel mantığı, görüntüyü oluşturan piksellerin birbirleriyle veya onların istatistiksel değerleriyle karşılaştırılması işlemine dayanmaktadır. Bu konuda bir çok yöntem geliştirilmiş ve bu yöntemler eğitilmiş ve eğitimsiz olmak üzere iki sınıfa ayrılmıştır. Ancak günümüze kadar geliştirilen yöntemlerin yeterince tatminkar sonuçlar vermemesi ve bazı yöntemlerin çok yavaş işlemesiyle birlikte bazılarının da çok büyük hafızaya sahip bilgisayar sistemlerine gereksinim duyması, bu konularla ilgilenenleri bu yöntemleri iyileştirme ve yeni yöntemler geliştirme çalışmalarına itmiştir.

Bu çalışmada, yukarıda sözü edilen amaçlar doğrultusunda eğitilmiş sınıflandırma yöntemleri incelenmiştir. Bu yöntemler, Çukurova Üniversitesi Fizik Bölümünde geliştirilen bir görüntü işleme programıyla koordinatı 175-34 olan LANDSAT-TM uydu verisinin 198x200 piksellik bölümü üzerinde uygulanmış ve sınıflandırma yöntemleri doğruluk ve işlem hızı açısından birbirleriyle karşılaştırılmıştır.

Çalışmanın bu bölümünü takip eden ikinci bölümünde, sınıflandırma ve doğruluk belirleme konularında Türkiye ve diğer ülkelerde yapılmış çalışmalardan bazıları özetlenmiştir. Üçüncü bölümde bu çalışmada izlenen yol, görüntü işleme tekniğinin temelleri ve eğitilmiş sınıflandırma yöntemleri açıklanmıştır. Dördüncü bölümde araştırma ve uygulamalar sonucunda elde edilen bilgiler sunulmuştur. Sonuç bölümünde ise eğitilmiş sınıflandırma yöntemleri, elde edilen bilgiler doğrultusunda karşılaştırılarak aralarındaki farklılıklar açıklanmıştır.

2. ÖNCEKİ ÇALIŞMALAR

1972 yılında ilk LANDSAT uydusunun yörüngesine yerleştirilmesinden itibaren yeryüzündeki nesneler hakkında veri toplanmaya başlanmış ve bu verileri değerlendirme çalışmaları yapılmıştır. Net görüntü ve doğru bilgi üretimi için yeni uydu sistemlerinin yanı sıra, çok sayıda görüntü işleme teknikleri de geliştirilmiştir. Bu tekniklerden en çok öneme sahip olan sınıflandırma işlemi konusunda, bugüne kadar bir çok yöntem geliştirilmiş ve bunların birbirleriyle karşılaştırılması araştırmaları süreklilik kazanmıştır. Geliştirilen yöntemler genellikle yer gerçeği çalışmalarıyla da desteklenmiştir. Aşağıda bu çalışmalardan bazıları özetlenmiştir.

İNCE (1986), Maksimum Benzerlik sınıflandırma yönteminin güvenilirliğini incelemiş ve bu amaçla En Yakın Komşu sınıflandırma yöntemiyle karşılaştırmıştır. Bu iki yöntemle sınıfladığı görüntüdeki her bir sınıfa ait piksel sayılarını ve işlem sürelerini değerlendirmiş ve hata oranlarını belirlemiştir. Maksimum Benzerlik yöntemi ile ilgili doğruluğun örnek sınıf verilerindeki istatistiksel değerlere önemli bir derecede bağımlı olduğunu ve En Yakın Komşu yönteminin daha iyi sonuçlar verdiğini ortaya koymuştur.

YEĞİNGİL (1987), eğitimsiz sınıflandırma işleminde kullanılan ve kümeleme adı verilen yöntemlerde sınıfların birbirleriyle karışma problemini en aza indirmek için yeni bir algoritma geliştirmiş ve bu algoritmayı Landsat TM verilerine uygulamıştır.

WANG (1990), Fuzzy eğitilmiş sınıflandırma yöntemini incelemek ve geliştirmek amacıyla yaptığı çalışmada, bu yöntemi Kanada'ya ait bir bölgenin 1978 tarihli Landsat MSS ve TM verisi üzerinde 7 ayrı sınıf için uygulamış ve sonuçları diğer eğitilmiş sınıflandırma yöntemlerinin sonuçlarıyla karşılaştırmıştır. Diğer yöntemlerle %86 civarında, Fuzzy yöntemiyle ise %91.21 oranında doğruluklar elde etmiştir. Ayrıca, bu araştırmanın, Fuzzy yönteminin uzaktan algılama tekniği ile elde edilen görüntüler için yeni bir yorumlama yöntemi olduğunu ortaya çıkardığını vurgulamaktadır.

FRANKLIN ve WILSON (1991), görüntü piksellerinin konumlarına dayanan yeni bir sınıflandırma yöntemi konusunda çalışmış ve bu yöntemin uygulanabileceği bir program hazırlamışlardır. Yöntemin doğruluğunu belirlemek amacıyla dağlık bir bölgeye (Yukon) ait veri setlerini bu yöntem ile sınıflandırmış ve sonuçta %56-%88 aralığında doğruluklar elde etmişlerdir.

TAKAGI ve KENMOCHI (1991), Landsat-5 TM uydusunun farklı mevsim ve yıllarına ait verilerini kullanarak, Japonya'daki Matsumoto adlı bir şehri 7 ayrı kategoride En Yakın Komşu yöntemi ile sınıflandırarak, sonuçların doğruluğu konusunda çalışmalar yapmışlardır. İyi sonuçlar alabilmek için, çok sayıda bantın (4, 6, multi-temporal) kombinasyonlarını kullanmış ve farklı sınıflar için farklı doğruluk oranları (%40-%90) elde etmişlerdir. Bu çalışmanın sonucunda, sınıflandırma işlemi doğruluğunun mevsimlere, bant sayısına, sınıflara ve Mahalanobis mesafesine bağlı olduğunu vurgulamışlardır.

ARAI (1992), TM verilerinin sınıflandırılması işleminde, Maksimum Benzerlik yöntemi için kullanılmak üzere seçilen örnek alanlarda, tayfsal olarak çok farklı değerlere sahip pikselleri temizleme yoluyla yöntemin doğruluğunu ve işlem hızını arttırmayı amaçlayan yeni bir algoritma geliştirmişlerdir. Bu algoritmayı, Kanada'nın Melfort ve Saskatchewan şehirlerindeki tarımsal alanlar üzerinde 7 tane farklı sınıf için uygulamışlar ve doğruluk oranını %5 - %11 aralığında (sınıflara göre değişiyor) arttırmışlardır.

MITİŞ (1992), eğitimsiz sınıflandırma işlemi için yeni bir algoritma geliştirmiş ve bu algoritmayı diğer eğitimsiz sınıflandırma yöntemleriyle karşılaştırmıştır. Bu yöntemleri, hazırladığı bir programla, GAP alanına ait bir bölgenin Landsat-5 TM verileri üzerinde toprak çeşitlerini sınıflandırmak amacıyla uygulamış ve yeni yöntemle doğruya çok yakın sonuçlar elde edildiğini vurgulamıştır.

WANG ve HOWARTH (1993), sınıflandırma işleminde belirsiz piksellerin yerleşecekleri sınıfların tahminini amaçlayan, yeni bir hata modeli geliştirmişler ve yaptıkları testler sonucunda, yeni modelin %95 oranında bir güvenilirliğe sahip olduğunu göstermişlerdir.

EMRAHOĞLU (1993), Adana iline ait Landsat-5 TM verilerini kullanarak, eğitilmiş sınıflandırma yöntemleri ve En Yakın Komşu yöntemi ile ilgili olarak geliştirdiği yeni algoritma arasında, doğruluk ve işlem hızları açısından karşılaştırma yapmıştır. Denemeler sonucunda yeni algoritmanın, diğer En Yakın Komşu yöntemlerine göre daha hızlı olduğunu ortaya koymuştur.

CONESE ve Ark. (1993), engebeli arazilerin sınıflandırılmasında, Maksimum Benzerlik yönteminin performansını ana bileşenler analizini kullanarak arttırmak amacıyla yeni bir algoritma geliştirmişler ve bu algoritmayı İtalya'nın kuzeyinde dağlık bir bölgenin TM verileri üzerinde uygulamışlardır.

INOMATA ve OGATA (1993), sınıflandırma işlemini birbirine komşu pikselleri içeren hücreler bazında yapabilen yeni bir algoritma geliştirmişlerdir. Bu algoritmanın karar kuralında ilk koşul olarak, hücreler ait histogramların simetrik olması yer almaktadır. Bu yöntem, birbirine alternatif iki teknikten oluşmakta (CDM-Cell Distance Method ve HOM-Histogram Overlay Method) ve ayrıca eğitimsiz sınıflandırma işlemine de uyarlanabilmektedir. Her iki tekniği de MOS-1/MESSR verileriyle Japonya'nın Hiroshima ve Kure şehirlerini çevreleyen denizdeki bulanıklığı araştırmada denemişler ve Maksimum Benzerlik yöntemi kadar iyi sonuçlar elde etmişlerdir. Ayrıca, CDM tekniğinin HOM tekniğinden çok küçük bir farkla daha doğru sonuçlar verdiğini, ancak daha yavaş olduğunu vurgulamışlardır.

FITZGERALD ve LEES (1994), sınıflandırmada doğruluğu ölçme işleminin güvenilirliğini araştırmak ve kullanılan hata modellerinin yanlış sonuçlar verebileceğini

göstermek amacıyla yaptıkları çalışmada, iki tane sınıflandırma yönteminin (Neural Network ve Decision Tree) aynı veri seti üzerindeki sınıflandırma işlemi sonuçlarına Kappa testi ile Toplam Doğruluk Yüzdesi testini uygulamışlardır. Bu iki hata modelini karşılaştırdıklarında Kappa testinin diğerinden çok daha güvenilir olduğunu ortaya çıkarmışlardır.

KARTIKEYAN ve Ark. (1994), biri alçak çözünürlüklü, diğeri ise yüksek çözünürlüklü veriler için olmak üzere iki ayrı eğitimli sınıflandırma yöntemi geliştirmiş ve bu yöntemleri, Maksimum Benzerlik yöntemi ile karşılaştırmışlardır. Bu iki yöntem, Maksimum Benzerlik yönteminde olduğu gibi, uzun matematiksel modellerden oluşmaktadır. Yöntemleri, IRS-1A uydusuna ait verilerin iki ayrı seti üzerinde uygulamış ve 1.yöntem için %91, 2.yöntem için %88 oranında doğruluklar elde etmişlerdir.

SHIH ve CHEN (1994), Fuzzy tekniğine dayanan ve iki aşamadan oluşan yeni bir eğitimsiz sınıflandırma algoritması geliştirmişlerdir. Bu algoritmayı, Landsat-TM uydusunun yedi tane bantından oluşan veri setinin 320x200 piksellik bölümü üzerinde uygulamışlar ve iyi olarak vurguladıkları doğruluklar elde etmişlerdir.

PEŞTEMALCI ve Ark. (1995), yeni bir eğitimli sınıflandırma yöntemi (Elipsoid) geliştirmiş ve yöntemin doğruluğunu araştırmak amacıyla Landsat-5 TM uydusunun 1991 yılına ait verilerini (3., 4. ve 5. bantlar) kullanarak Adana ilindeki buğday ve arpa alanlarını, %15 oranında bir hatayla 218.000 hektar olarak belirlemişlerdir.

3. MATERYAL VE METOD

3.1. Materyal

Bu araştırmada kullanılan materyaller, aşağıda maddeler halinde sıralanmıştır:

i. Eğitimli sınıflandırma yöntemlerinin incelenmesi, karşılaştırılması ve yeni bir sınıflandırma yöntemi geliştirilmesi üzerine çalışmalar yapılırken, daha önce üzerinde çalışmalar yapılarak yer gerçeği bilinen ve Adana il sınırları içerisinde bulunan bir bölgeye ait 27.03.1992 tarihli, yörünge ve satır numarası 175-34 olan LANDSAT-TM uydusunun 3., 4. ve 5. bantlardaki verileri kullanılmıştır. Bu verilerden faydalanarak işlenmeden (ham) oluşturulan görüntüler resim 1 'de verilmiştir.

ii. Yer gerçeğinin belirlenebilmesi için yukarıda sözü edilen veri setinin kapsadığı alanda hedef olarak seçilen bölgenin, Devlet Su İşlerinden (DSİ) elde edilen haritası (Şekil 2) (Emrahoğlu, 1993 'den).

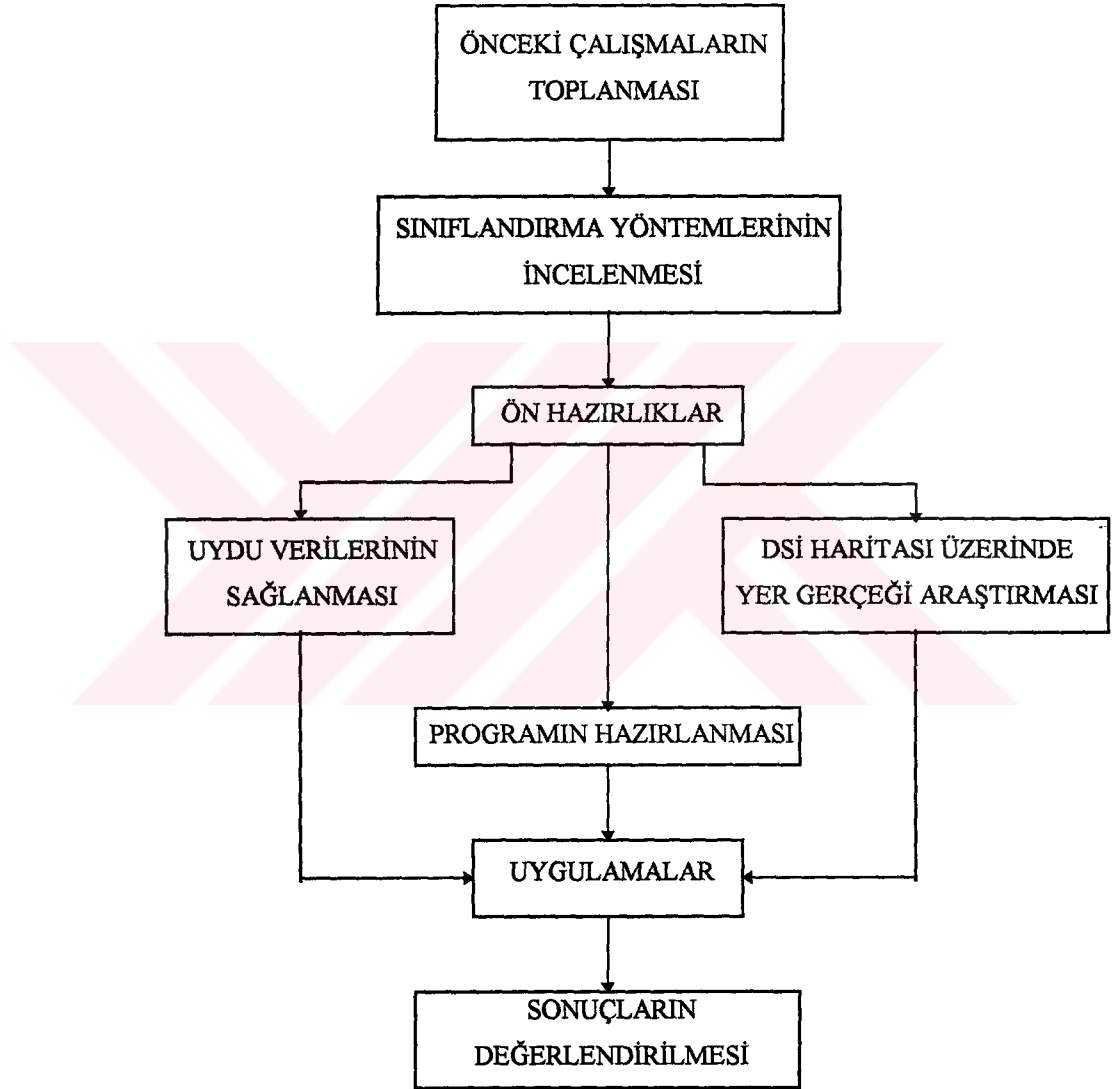
iii. Temel birimleri 486DX2-66 Mikroişlemci, 4 MB Ana Bellek, 540 MB Sabit disk, 14" 0.28 Renkli Ekran ve 1 MB bellekli (SVGA) grafik kartı olan bir mikrobilgisayar.

iv. Quick Basic programlama dilinde hazırlanan, sınıflandırma yöntemlerinin uygulandığı görüntü işleme programı (Bu programın listesi EK-1 'de verilmiştir).

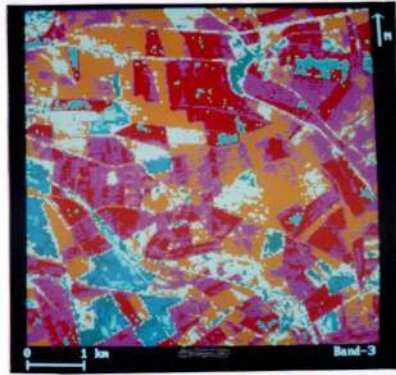
3.2. Metod

Bu araştırmanın ana hatlarını gösteren akış şeması şekil 1 'de verilmiş olup, başlıca şu adımlar izlenmiştir:

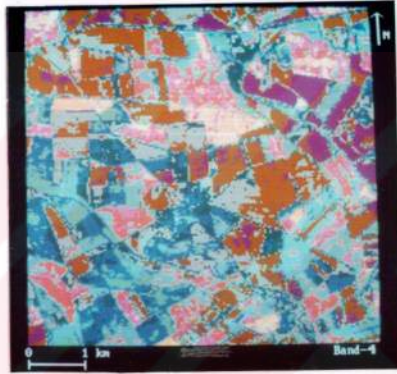
- DSİ haritası üzerinde yer gerçeği çalışmaları,
- Sınıflandırılacak görüntüde sınıf örneklerinin belirlenmesi,
- Görüntü üzerinde sınıflandırma yöntemlerinin uygulanması,
- Sınıflandırma işlemlerinin denetlenmesi,
- Doğruluk tablolarının hazırlanması,
- Sınıflandırma yöntemlerinin doğruluk ve işlem hızı açısından karşılaştırılması,
- Sonuçların değerlendirilmesi.



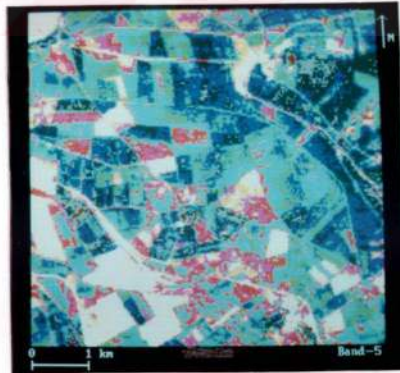
Şekil 1. Araştırmada izlenen akış şeması.



3. Bant



4. Bant



5. Bant

Resim 1. Çalışılan alanın Landsat TM uydu görüntüleri (3., 4. ve 5. Bantlar).



Şekil 2. Harita üzerinde yer gerçeği çalışması yapılan alanlar.

3.3. Sayısal Görüntü İşleme Konusunda Temel Prensipler

3.3.1. Görüntü

Günlük yaşamda resim de diyebileceğimiz “görüntü” (image) en genel anlamıyla iki boyutlu bir sinyal kayıdır. Fotoğraf ya da televizyon ekranı gibi gözle görünen bir biçimde olabileceği gibi, manyetik bantta yazılı bir kayıt veya bilgisayar belleğinde duran sayısal değerler biçiminde de olabilir (İnce, 1986).

3.3.1.1. Sayısal Görüntü İşleme

Sayısal görüntü işleme (Digital image processing), bir teleskopla gözlenen yıldızların parlaklığını tanımlamada, bir mikroskop görüntüsündeki virüsün yapısını tanımlamada ve yapay uydular aracılığıyla elde edilen yeryüzü görüntülerinden doğruluğu çok yüksek olan haritaların oluşturulmasında kullanılmaktadır. Genel olarak sayısal görüntü işlemenin amacı, görüntüyü bazı metodlarla geliştirmek veya ondan bilgi almaktır. Bu konudaki en genel uygulamalar:

- Görüntüdeki bulanıklıkları yok etmek,
- Görüntünün kalitesini artırmak,
- Görüntüdeki parlaklık veya başka görsel özellikleri belirlemek,
- Görüntüyü bölgelere ayırmak (sınıflandırmak)
- Görüntüyü büyütmek, küçültmek veya döndürmek,
- Görüntüdeki eğrilik ve çarpıklıkları yok etmek,
- Görüntüyü depolayabilmek veya başka bir biçime dönüştürebilmek için kodlamak.

şeklinde sıralanabilir (Niblack, 1986).

3.3.1.2. Görüntünün Sayısallaştırılması (Digitization)

Amaç ne olursa olsun, görüntülerin bilgisayar sistemleriyle işlenebilmesi için, bunların bilgisayara uygun bilgiye dönüştürülmesi gerekir. Bu dönüştürme işlemine görüntünün sayısallaştırılması denir. Bir çok uygulamada, özellikle uzaktan algılamada sayısallaştırma, bir uçak veya uydu içindeki çok banlı satır tarayıcıların (multispectral scanner) veriyi kaydetmesi esnasında yapılmaktadır.

Bu olaya örnek olarak, önce siyah-beyaz bir görüntünün sayısallaştırılması ele alınsın. İlk adımda, görüntü karelere bölünür (elek, ızgara gibi). İkinci adımda ise, görüntüdeki her bir karenin daha çok siyah mı veya beyaz mı olduğu kontrol edilir. Bu durumda her bir kare tamamıyla siyah veya beyaz olarak tanımlanabilir. Son olarak piksel (Piksel, bir resmin en küçük elemanı olarak tanımlanabilir) denilen bu siyah veya beyaz kareler sırasıyla (0) ve (1) olarak kodlanır. Böylece, orijinal resmin ikili (binary) sayısal sunumu elde edilir.

Siyah-beyaz bir görüntünün sayısallaştırılmasının kalitesi, karelerin (piksel) büyüklüğüne bağlıdır. Piksellerin yatay durumda sıralanmış haline bir görüntü satırı, düşey durumda sıralanmış haline ise görüntünün bir kolonu denir. Satırlar yukarıdan aşağıya doğru, kolonlar ise soldan sağa artacak şekilde numaralandırılır.

Bir matematiksel notasyonla, bir ikili görüntüyü karakterize edebiliriz. Bu durumda,

x, y : Bir pikselin konumu (x : satır no, y : kolon no)

$s(x,y)$: (x,y) konumundaki pikselin ikili değeri (0 - siyah, 1 - beyaz) olmak üzere, sayısallaştırılmış bir görüntünün ikili matrisi $S=s(x,y)$ dir.

Bir gri tonlu görüntü, iki gri tondan daha fazla tona sahiptir. Bu yüzden, daha fazla rengin kodlanabilmesi gerekir. Örneğin, 8 bitlik bir bilgisayarda $2^8=256$ olduğundan, yani 8 tane bit (0 ve 1) ile 256 farklı değer ($G=\{0,1,2,...,255\}$) temsil edilebileceğinden, sayısallaştırma işleminde 256 tane renk kodlanabilir.

Renkli bir görüntünün sayısallaştırılmasında kırmızı (red), yeşil (green) ve mavi (blue) filtreler kullanılarak bir renk ayrımı oluşturulmaktadır. Bu durumda sayısallaştırılmış renkli bir görüntünün bir pikseli, üç boyutlu bir vektördür.

$$s(x,y,n)=(r,g,b)$$

Burada, r , ortalama kırmızı yoğunluğunu, g , ortalama yeşil yoğunluğunu, ve b ise ortalama mavi yoğunluğunu gösterir. Örneğin, (127,0,0) pikseli tamamıyla orta yoğunluklu kırmızı rengi, (0,0,0) kombinasyonu siyahı ve (255,255,255) beyazı gösterir.

Çok bantlı bir tarayıcı N _boyutlu pikseller üretir ($g_0, g_1, \dots, g_{N-1}$). Burada, g_i , i . tayfsal banttaki yoğunluk kodunu gösterir. Daha önce belirtilen matematiksel modele göre, çok bantlı bir görüntü üç boyutlu bir matris olarak yazılacaktır.

$$S=s(x,y,n)$$

Burada, (x,y) yersel koordinatlar, n ise tayfsal bantlardır. Bazı hallerde, N _boyutlu pikselin bileşenleri tayfsal olarak gösterilmez fakat keyfi bazı bilgiler sunabilir. Örneğin, sayısal saha modelinin rakımının kodlanmış olabilir. Bu tür bir sayısal görüntüye çok kanallı (multichannel) görüntü denir. Çok bantlı görüntüler bazı hallerde, çok kanallı görüntülerin özel halleridir (Haberacker, 1984).

Çok kanallı görüntüler farklı zamanlarda algılanabilir ve bu zaman sırasını da içeren görüntü dört boyutlu matris ile gösterilir.

$$S=s(x,y,n,t)$$

3.3.1.3. Sayısallaştırılan Görüntü Verilerinin Kaydedilmesi

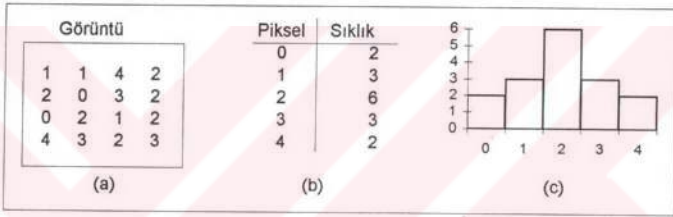
Elde edilen S değerleri aynı adresle (aynı (x,y) değerleriyle) bir disk dosyasına üç değişik formatta kaydedilebilmektedir. Uygulanacak işleme göre bu formatlardan biri tercih edilir. Bu formatlar kendi içinde de değişiklik gösterebilir, ancak görüntü dosyalarındaki veri sıralanışları şöyledir:

- i. Bant-Bant : Önce birinci bantın birinci satırı, sonra birinci bantın ikinci satırı, ..., birinci bantın sonuncu satırı, ikinci bantın birinci satırı, sonra ikinci bantın ikinci satırı, ..., ikinci bantın sonuncu satırı, ... şeklinde devam eder (BSQ-Band Sequential).
- ii. Satır-Satır : Birinci bantın birinci satırı, ikinci bantın birinci satırı, ..., son bantın birinci satırı, birinci bantın ikinci satırı, ikinci bantın ikinci satırı, ..., son bantın ikinci satırı, ... şeklinde devam eder (BIL- Band interleaved by line).
- iii. Piksel-Piksel : Birinci piksel (birinci banttan sonuncu banta), ikinci piksel (birinci banttan sonuncu banta), ... , birinci satırın son pikseli, ikinci satırın ilk pikseli, (birinci banttan sonuncu banta), ... şeklinde devam eder (BIP- Band interleaved by piksel) (İnce, 1986).

3.3.2. Görüntü İşlemede Kullanılan İstatistiksel Özellikler

3.3.2.1. Görüntü Histogramı

$S=s(x,y)$, $G=\{0,1,2, \dots ,255\}$ kümesindeki değerlere sahip piksellerden oluşan bir görüntü olsun. S 'nin histogramı, G kümesindeki her bir değere sahip piksellerden kaç tane olduğunu gösteren bir tablodur (Şekil 3). Histogram $h(s)$ ile gösterilir ve genellikle grafiksel olarak belirtilir. Kısacası histogram, bir görüntüdeki piksel değerlerinin dağılımının bir ölçüsüdür.



Şekil 3. a) Sembolik bir görüntüye ait veri seti, b) Veri setindeki piksel değerleri ve bu piksellerin sıklığı, c) Sembolik görüntünün histogramı.

Buna göre, histogramı hesaplanan bir görüntüdeki toplam piksel sayısı aynı zamanda,

$$q = \sum_{s=0}^{255} h(s) \quad (3.1)$$

şeklinde de hesaplanabilir. Eğer histogram normalize edilirse:

$$H(s) = \frac{h(s)}{q} \quad (3.2)$$

bulunur. Buradaki $H(s)$, istatistiksel olasılık yoğunluk fonksiyonuna benzer ve bir pikselin s değerine sahip olma olasılığı olarak da düşünülebilir. Bu durumda,

$$\sum_{s=0}^{255} H(s) = 1 \quad (3.3)$$

olduğu görülür (Niblack, 1986).

3.3.2.2. Ortalama Değer ve Standart Sapma

Bir görüntüdeki piksellerin dağılımını bu piksellerin ortalaması ve standart sapması bir yaklaşım olarak gösterebilir. q görüntüdeki toplam piksel sayısı olmak üzere, ortalama

$$\mu = \frac{1}{q} \sum_{x,y} s(x,y) \quad (3.4)$$

standart sapma ise,

$$\sigma = \sqrt{\frac{1}{q-1} \sum_{x,y} (s(x,y) - \mu)^2} \quad (3.5)$$

şeklinde ifade edilir.

3.3.3. Sınıflandırma

Sınıflandırma, en genel anlamıyla, çok çeşitli nesnelerin rasgele bir araya gelerek oluşturduğu setin, aynı veya benzer özelliklere sahip olan elemanlarını ayrı ayrı gruplandırarak daha belirgin hale getirmektir. Uzaktan algılama olayında, bu elemanlar bir görüntüyü oluşturan piksellerdir. Bu işlemin en genel uygulaması, yeryüzünün belli bir bölgesindeki ekili alanlar, su ve orman gibi değişik yeryüzü örtülerini sınıflamaktır.

Sınıflandırma, aynı zamanda bir etiketleme problemi olarak da düşünülebilir. Bir bölgedeki her nesne bir sınıf adıyla (veya numarasıyla) etiketlenilirse, bölgenin detayları ortaya çıkar ve böylece ilgili karar işlemleri kolaylaşır.

Eğitilmiş ve eğitimsiz olmak üzere iki temel sınıflandırma yöntemi vardır. Eğitilmiş sınıflandırma yönteminde, araştırmacı daha önceden bilgi sahibi olduğu sınıfları temsil eden bölgelerden örnekler alarak ve sınıflandırılacak görüntünün her pikselini bu örneklerle karşılaştırarak veya bu örneklerle bağımlı olasılık fonksiyonlarını kullanarak sınıflandırma işlemini yapar. Eğitimsiz sınıflandırma yönteminde ise, elde bilinen bir örnek olmadan piksel uzayında bulunan her bir kümenin bir sınıfa karşılık geldiği varsayılarak işlem yapılır. Buradaki problem, hangi kümenin hangi fiziksel yapıya (su, buğday, vb.) karşılık geldiğinin bilinmemesidir. Sınıflandırma işlemini daha iyi açıklayabilmek için şöyle bir örnek verilebilir: Bilgisayar ekranında sadece su ve karayı

içeren kıyasal bir bölgenin bir bantta görüntüsü olsun. Bu görüntünün her bir pikselini ya suya (sınıf-1) ya da karaya (sınıf-2) atıttır şeklinde bir karar vererek sınıflandırmak isteyelim. Sonuçta, her bir pikselinin kodu 1 veya 2 olan bir görüntü (sınıf haritası) elde edilecektir.

Her bir pikseli sınıflardan birine atayabilmek için bir karar kuralına ihtiyaç vardır. Örneğin, saçma da olsa “Görüntünün sol tarafındaki bütün pikseller suyu temsil eden sınıfa, sağındakiler ise karayı temsil eden sınıfa atansın” diye bir kural ortaya atılabilir.

Bu keyfi kural, doğal olarak doğru bir sonuç vermeyecektir. Ama bu görüntünün, toprak olarak bilinen bir kaç bölgesinin ve su olarak bilinen bir kaç bölgesinin piksel değerlerine bakılırsa, o zaman “Değeri 18’den küçük olan bütün pikseller suya, diğerleri ise karaya atıttır” şeklinde daha güvenilir bir kuraldan bahsedilebilir (Niblack,1986).

Gerçekte de, eğitilmiş olarak adlandırılan sınıflandırma yöntemlerindeki karar kuralları için genellikle bir görüntüdeki sınıfları temsil edecek örnek piksel gruplarının ayrı ayrı histogramlarına, ortalamalarına ve varyans-kovaryans gibi istatistiksel bilgilerine ihtiyaç vardır. Bu bilgilerin elde edilmesi ve uygulanması aşağıdaki eğitilmiş sınıflandırma yöntemlerinde açıklanmaktadır.

3.3.3.1. Eğitilmiş Sınıflandırma Yöntemleri

3.3.3.1.1. Maksimum Benzerlik Sınıflandırması

Bu yöntem (Maximum Likelihood Classification) araştırma çevrelerinde en yaygın olarak kullanılan ve hemen hemen bütün görüntü işleme yazılımlarında bulunan bir yöntemdir. Bu yöntemi açıklamak için yine yukarıdaki örnek ele alınsın.

Bütün görüntünün su ve kara olarak bilinen birkaç örnek alanındaki piksellerin histogramları ($h_k(s)$ ve $h_k(s)$) hesaplanırsa, görüntünün her pikseli “s değerine sahip piksel, eğer $h_k(s) > h_k(s)$ ise karayı temsil eden sınıfa, değilse suyu temsil eden sınıfa dahil edilir” kuralına göre sınıflandırılabilir.

Bu kural, yanında bir problemi de getirmektedir. Eğer karayı temsil eden örnek alanlarda diğerinden çok daha fazla piksel varsa, onun histogramı da genellikle daha büyük, dolayısıyla sınıflandırmadaki ağırlığı daha fazla olacaktır. Eğer histogramlar

normalize edilirse ($H_k(s)=h_k(s)/q_k$ ve $H_s(s)=h_s(s)/q_s$), bu problem daha açık bir şekilde görülür. Buna göre, görüntüdeki tüm piksellerin örneğin %70'i karaya, %30'u ise suya aitmiş gibi görünecektir. Bunun anlamı, eğer görüntüden rasgele bir piksel seçilirse, bu pikselin karayı temsil eden sınıfa ait olma olasılığı 0.7, suya ait olma olasılığı ise 0.3'tür. Bu değerlere sırasıyla a_k ve a_s ilk olasılıkları denir. Buna göre, görüntünün her pikseli “s değerine sahip piksel, eğer $a_k H_k(s) > a_s H_s(s)$ ise karayı temsil eden sınıfa, değilse suyu temsil eden sınıfa dahil edilir” kuralına göre sınıflandırılabilir.

Bu, Maksimum Benzerlik kuralının temelidir. Bu kural, her bir sınıf başına bir histogram kullanarak daha çok sınıf için ve n boyutlu histogramlar kullanarak daha çok bant için genişletilebilir ve yine bu temele bağlı olarak aşağıdaki dönüşümlerle iki tane istatistiksel sonuç elde edilebilir.

$P(i)$: Bir pikselin i sınıfından olabilesinin ilk olasılığı,

$P(s|i)$: i sınıfından bir pikselin s değerine sahip olma olasılığı,

$P(i|s)$: s değerine sahip bir pikselin i sınıfına ait olma olasılığı,

$P(s)$: Tüm i değerleri için $P(s|i)$ 'lerin toplamı

olmak üzere bu terimlerle tanımlanan Bayes Formülü ilk sonucu verir:

$$P(i|s) = \frac{P(s|i)P(i)}{P(s)} \quad (3.6)$$

Bu formülle, s değerine sahip pikselin i sınıfına ait olma olasılığı, buna zıt olasılıktan, yani i sınıfından bir pikselin s değerine sahip olma olasılığından hesaplanır.

İkinci sonuç ise sınıflandırma işleminde kullanacağımız karar kuralıdır. Bayes karar kuralı olarak adlandırılır ve yine aynı terimleri içerir. “s değerine sahip bir pikselin i sınıfından olma olasılığı $P(i|s)$ ise pikselin ait olabileceği en iyi sınıf, her bir sınıf için hesaplanan $P(i|s)$ 'lerden maksimum olanıdır.”

Bu sınıflandırma yönteminin uygulanmasında aşağıdaki adımlar sırayla izlenir:

1. Her bir aday sınıf için örnek alanlar seçilir. Bunlar sınıfı bilinen pikselleri içeren alanlardır. Her bir sınıf için birden fazla örnek alan seçilmesi daha iyi sonuçlar verecektir.
2. Her bir aday sınıf için seçilen örnek alanlardaki verilerin n boyutlu histogramları hesaplanır ve histogramlar normalize edilir. Bunlar i sınıfından bir pikselin s değerine

sahip olma olasılığını veren olasılık yoğunluk fonksiyonlarının ($P(s|i)$) tahmini için kullanılır. Her bir i sınıfı için bir tane $P(s)$ vardır.

3. İlk olasılıklar ($P(i)$) tahmin edilir ve bunlar $P(s|i)$ 'lerin hesaplanması için kullanılır.

4. Bağntı (3.6) ile görüntüdeki her bir piksel, her bir i sınıfı için hesaplanarak sınıflandırılır. Pratikte, $P(s)$ her i sınıfı için aynı olduğundan formülde atılır ve formül

$$g(i|s)=P(s|i)P(i) \quad (3.7)$$

haline dönüşür. Ve buna göre, her piksel $g(i|s)$ 'nin maksimum olduğu sınıfa konur. $g(i|s)$ terimi normalde s 'nin bir fonksiyonu olarak düşünülür ve bir olasılıktır. Bununla birlikte, i 'nin bir fonksiyonu olarak düşünüldüğü zaman, benzerlik fonksiyonu olarak adlandırılır ve bu düşünce yönteme Maksimum Benzerlik Sınıflandırması adı verilir (Niblack,1986).

Yukarıdaki adımlar temel bir yaklaşımdır. Ancak bu olayı bir bilgisayar programıyla yürütmeye bellek sorunu ile karşılaşılır. 2. adımda $P(s|i)$, örnek verilerin n boyutlu histogramı normalize edilerek hesaplanıyordu. Eğer, i tane sınıf, n tane bant, her bantta da m tane piksel varsa, bu verilerle $P(s|i)$ 'lerin hesaplanabilmesi için belleğin ixm^n büyüklüğünde bir tabloyu saklayabilmesi gerekir. Örneğin, 4 adet bant, her bantta 0'dan 255'e kadar değerlere sahip pikseller ve 8 tane sınıf varsa, bilgisayar belleğinin $8 \times 256^4 = 34.36$ Milyar Byte'dan büyük olması gerekir.

Bu sakıncayı gidermek için ilgilenilen herhangi bir sınıfın olasılık fonksiyonuna bir normal (veya Gauss) yoğunluk fonksiyonu ile yaklaşılabileceği varsayılır. i . sınıf için normal yoğunluk fonksiyonu tek boyutta (tek bant için):

$$P(s|i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{1}{2}\frac{(s-\mu_i)^2}{\sigma_i^2}\right] \quad (3.8)$$

olarak yazılır. Burada μ_i , i . sınıf için seçilen örnek piksellerin ortalama değeri, σ_i ise yine bu piksellerin varyansı olmak üzere

$$\mu_i = \frac{1}{q_i} \sum_{j=1}^{q_i} s_j \quad (3.9) \quad \text{ve} \quad \sigma_i^2 = \frac{1}{q_i - 1} \sum_{j=1}^{q_i} (s_j - \mu_i)^2 \quad (3.10)$$

dir.

Bu varsayım yapıldıktan sonra, bilgisayar belleğinde histogramın tümü yerine sadece her i sınıfı için seçilen örnek piksellerin (Şekil 4) ortalamasını ve varyansını bulundurmak yeterlidir. Sınıflandırma esnasında bu verilerle bir pikselin her aday sınıf için olasılık fonksiyonu hesaplanabilir ve bu piksel, $P(s|i)$ fonksiyonunun maksimum olduğu sınıfa dahil edilir. Ayrıca bu fonksiyonun matris gösteriminden faydalanarak, n boyut (n bant) için genel formu türetilir. Eğer

$$S = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \quad U_i = \begin{bmatrix} \mu_{i1} \\ \mu_{i2} \\ \vdots \\ \mu_{in} \end{bmatrix} \quad \Sigma_i = \begin{bmatrix} \sigma_{i11} & \dots & \sigma_{i12} & \dots & \sigma_{i1n} \\ \sigma_{i21} & \dots & \sigma_{i22} & \dots & \sigma_{i2n} \\ \vdots & & \vdots & & \vdots \\ \sigma_{in1} & \dots & \sigma_{in2} & \dots & \sigma_{inn} \end{bmatrix}$$

ifadelerine sırasıyla, veri vektörü, i . sınıfın ortalama vektörü ve i . sınıfın kovaryans matrisi denirse, n boyutlu normal yoğunluk fonksiyonu

$$P(S|i) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (S - U_i)^T \Sigma_i^{-1} (S - U_i) \right] \quad (3.11)$$

şeklinde yazılabilir. Burada, $|\Sigma_i|$, Σ_i kovaryans matrisinin determinanı, Σ_i^{-1} , Σ_i kovaryans matrisinin tersidir. Bu fonksiyon için gerekli olan her sınıfın ortalama vektörü ve kovaryans matrisi,

$$\mu_{ij} = \frac{1}{q_i} \sum_{l=1}^{q_i} s_{jl} \quad (3.12) \quad \text{ve} \quad \sigma_{ijk} = \frac{1}{q_i - 1} \sum_{l=1}^{q_i} (s_{jl} - \mu_{ij})(s_{kl} - \mu_{ik}) \quad (3.13)$$

$$(j=1,2,\dots,n, \quad k=1,2,\dots,n \text{ ve } l=1,2,\dots,q_i)$$

bağıntılarıyla hesaplanır. Burada kovaryans matrisin determinantının sıfır veya negatif olduğu durumlarda dikkat edilmelidir. Bilgisayarda daha kolay hesaplanabilirliği açısından, n boyutlu normal yoğunluk fonksiyonu aşağıdaki hale dönüştürülebilir:

$$g_i(S) = \log_e P(i) - \frac{1}{2} \log_e |\Sigma_i| - \frac{1}{2} (S - U_i)^T \Sigma_i^{-1} (S - U_i) \quad (3.14)$$

Bu durumda karar kuralı, "Piksel, $g_i(S)$ 'in minimum olduğu sınıfa dahil edilir" şeklinde değişir.



Şekil 4. Görüntünün 3 banttaki verileri ve her bantın aynı adresinden sembolik olarak seçilen sınıf örnekleri.

3.3.3.1.2. Minimum Mesafe Sınıflandırması (Minimum Distance Classification)

Bir önceki konuda anlatılan Maksimum Benzerlik yöntemi çok bantlı görüntü verileri için en çok kullanılan yöntem olmasına rağmen, uygulama esnasında uğraştırıcı olmakta ve aynı zamanda sınıf dağılımında Gauss yaklaşımını kullanmak bazen problemlere yol açmaktadır.

Bu yöntem alternatif olan Minimum Mesafe yönteminde, her sınıfın yalnızca ortalama vektörü bulunup, sınıfların kovaryans matrislerinin eşit olduğu varsayılır. (İnce, 1986). Bunun için, μ_i test alanından seçilen i . sınıfı temsil eden herhangi bir banta ait örnek alandaki piksellerin ortalama değeri olmak üzere (N_{bant} probleminde, μ_i ,

N _boyutlu uzayda bir vektör), görüntünün s değerine sahip pikselinin her i sınıfı için μ_i 'ye mesafesi:

$$d(s, \mu_i) = \sqrt{(s_1 - \mu_{i1})^2 + (s_2 - \mu_{i2})^2 + \dots + (s_n - \mu_{in})^2} \quad (3.15)$$

bağıntısıyla (Euclid Mesafesi) veya

$$d(s, \mu_i) = |s_1 - \mu_{i1}| + |s_2 - \mu_{i2}| + \dots + |s_n - \mu_{in}| \quad (3.16)$$

bağıntısı ile hesaplanır. Bu denklemlerdeki s_n , sınıflanacak pikselin n . banttaki değeri, μ_{in} ise n .banttaki ortalama değerdir. Sınıflandırma esnasında ise, “ s değerine sahip piksel, mesafesinin ($d(s, \mu_i)$) en küçük olduğu sınıfa dahil edilir” (Niblack, 1986). Bu yöntem Maksimum Benzerlik yöntemine göre daha hızlı, fakat hata oranı daha yüksektir.

3.3.3.1.3. Kutu Sınıflandırması (Box veya Parallelepiped)

Bu yöntemde, i sınıfı için seçilen örnek alanlardaki piksellerin yalnızca her bantta ortalama ve standart sapma değerlerine ihtiyaç duyulur. Daha önceden belirtildiği gibi çok bantlı bir görüntünün N _boyutlu ortalama vektörü,

$$U_i = (\mu_{i1}, \mu_{i2}, \dots, \mu_{in})$$

ve standart sapma ise,

$$\sigma_i = (\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{in}) \quad \text{idi.}$$

Burada, μ_{in} , i . sınıf için seçilen örnek alanın n . banttaki ortalaması, σ_{in} , standart sapması ve $s(x,y,n)$ de sınıflanacak görüntü pikselinin n . banttaki değeri olmak üzere, piksel ancak ve ancak

$$\mu_{in} - \sigma_{in} \leq s(x,y,n) \leq \mu_{in} + \sigma_{in}$$

ise i . sınıfa dahil edilir.

Piksel bu aralıkta değilse, hiçbir sınıfa dahil edilmez. Ancak standart sapmanın iki veya üç katı (2σ veya 3σ) alınarak aralık genişletilebilir.

Bu yöntem “kutu” adını, 3 bantlı bir görüntü için hesaplanan μ_{i1} , μ_{i2} ve μ_{i3} değerlerinin, bir kutunun birbirine dik üç kenarının uzunlukları olarak kabul edilmesinden dolayı alır. Bu durumda,

$$\mu_{i1} \pm \sigma_{i1}, \mu_{i2} \pm \sigma_{i2} \text{ ve } \mu_{i3} \pm \sigma_{i3}$$

değerleri her bir paralelin sınırlarının konumunu vermektedir. Sınıflandırma esnasında ise, piksel değerinin bu sınırlar aralığında olup olmadığına, yani pikselin bu kutunun içerisine düşüp düşmediğine bakılır.

Bu yöntemde, pikselin birden fazla sınıfı temsil eden kutu içerisine düşme olasılığı bulunmaktadır. Bu problem, pikselin karşılık geldiği ilk sınıfa konmasıyla geçiştirilir. Veya bu sınıflar için ayrıca “En Küçük Mesafe” yönteminin uygulanması en iyi çözüm olabilir.

3.3.3.1.4. Elipsoidal Sınıflandırma Yöntemi

Bu yöntem ise, bir önceki konuda anlatılan Kutu Sınıflandırması esasına dayanmaktadır. Kutu köşelerinin ortalamasının uzağında kaldığı düşünülerek, bunların yuvarlatılması sonucunda oluşan elipsoidin, sınıflanacak görüntü pikselini kapsayıp kapsamadığına bakılır.

Bu işlem için yine her bir i sınıfı için seçilen örnek alanlardaki piksellerin üç bantta ortalama değerleri (μ_{in}) ve varyansları (σ_{in}^2) bulunur. Sınıflandırma esnasında ise, $s(x,y,n)$ görüntü pikseli ancak ve ancak

$$\left[\sum_{n=1}^3 \frac{(s(x,y,n) - \mu_{in})^2}{(R\sigma_{in})^2} \right] \leq 1 \quad (3.17)$$

ise i . sınıfa dahil edilir. Burada R , güven aralığıdır ($R=1-4$).

3.3.3.1.5. En Yakın Komşu Yöntemi (Nearest Neighbor)

Bu yöntemde sınıfları temsil eden örnek alanların istatistiksel değerleri kullanılmaz. Örnek alanlardaki pikseller, doğrudan görüntü pikselini sınıflamak için kullanılırlar. Daha açık anlatmak için örnek olarak 1 banttaki görüntü ele alınırsa, sınıflanacak görüntü pikseli, i sınıfı için seçilen örnek alanın bütün pikselleriyle karşılaştırılarak kendisine en yakın pikselin mesafesi bulunur. Aynı işlem aynı pikselle diğer sınıfları temsil eden örnek alanların pikselleriyle tekrar edilir. İşlem sonucunda, en küçük mesafe hangi sınıftan geldiyse, piksel o sınıfa dahil edilir. $s(x,y)$, sınıflanacak görüntü pikseli, q , i . sınıf için

seçilen örnek alandaki toplam piksel sayısı ve p_{ij} , i . sınıfa ait j . piksel ($j=1,2, \dots, q$) olmak üzere, $s(x,y)$ 'nin p_{ij} 'ye mesafesi

$$\text{Dist}_i(j) = |s(x,y) - p_{ij}| \quad (3.18)$$

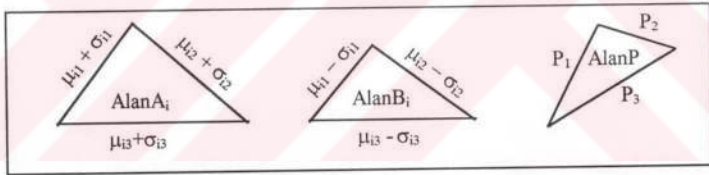
denklemleri ile bulunur. n bantlı görüntü için denklem $k=1,2, \dots, n$ olmak üzere

$$\text{Dist}_i(j) = \sum_{k=1}^n |s(x,y,k) - p_{ij}(k)| \quad (3.19)$$

şekline dönüşür.

3.3.3.1.6. Düşünülen Yeni Yöntem

Bu yöntemde, i sınıfı için seçilen örnek alanlardaki piksellerin yalnızca üç bantta ortalama ($\mu_{i1}, \mu_{i2}, \mu_{i3}$) ve standart sapma ($\sigma_{i1}, \sigma_{i2}, \sigma_{i3}$) değerleri gereksinim vardır. Bulunan bu değerlerin, iki tane üçgenin kenar uzunlukları olduğu (Şekil 5) varsayıp, bu üçgenlerin alanları aşağıdaki denklemlerle hesaplanır.



Şekil 5. Piksellerin ortalama ve standart sapma değerlerinin kenarlarını oluşturduğu üçgenler.

$$\text{Çev}A_i = ((\mu_{i1} + \sigma_{i1}) + (\mu_{i2} + \sigma_{i2}) + (\mu_{i3} + \sigma_{i3})) / 2 \quad (3.20)$$

$$\text{Alan}A_i = \sqrt{\text{Çev}A_i \cdot (\text{Çev}A_i - (\mu_{i1} + \sigma_{i1})) \cdot (\text{Çev}A_i - (\mu_{i2} + \sigma_{i2})) \cdot (\text{Çev}A_i - (\mu_{i3} + \sigma_{i3}))} \quad (3.21)$$

$$\text{Çev}B_i = ((\mu_{i1} - \sigma_{i1}) + (\mu_{i2} - \sigma_{i2}) + (\mu_{i3} - \sigma_{i3})) / 2 \quad (3.22)$$

$$(3.23)$$

$$\text{Alan}B_i = \sqrt{\text{Çev}B_i \cdot (\text{Çev}B_i - (\mu_{i1} - \sigma_{i1})) \cdot (\text{Çev}B_i - (\mu_{i2} - \sigma_{i2})) \cdot (\text{Çev}B_i - (\mu_{i3} - \sigma_{i3}))}$$

Sınıflandırma esnasında, her üç bantın aynı adresinden (satır, sütun) birer piksel alınarak (sınıflanacak görüntü pikseli) bunların değerleriyle aynı şekilde bir üçgen alanı hesaplanır.

$$\zeta_{evP} = (P_1 + P_2 + P_3)/2 \quad (3.24)$$

$$AlanP = \sqrt{(\zeta_{evP} \cdot (\zeta_{evP} - P_1) \cdot (\zeta_{evP} - P_2) \cdot (\zeta_{evP} - P_3))} \quad (3.25)$$

Karar kuralı ise, “Eğer $AlanP$, i . sınıf için hesaplanan $AlanA_i$ ve $AlanB_i$ aralığındaysa ($AlanB_i \leq AlanP \leq AlanA_i$) piksel i . sınıfa dahil edilir” şeklindedir.

Buradaki sakınca, $AlanP$ 'nin diğer sınıflar için bulunan $AlanA$ ve $AlanB$ aralıklarına da girme olasılığıdır. Bu durumda, $AlanP$ 'nin girdiği sınıflar için ayrı ayrı hesaplanan $AlanA$ ve $AlanB$ 'lerin ortalamalarına mesafeleri aşağıda verilen formülle ölçülür ve piksel, en küçük mesafeyi veren sınıfa dahil edilir.

$$Dist_i = |(AlanA_i + AlanB_i)/2 - AlanP| \quad (3.26)$$

4. ARAŞTIRMA BULGULARI

Uydu verileri görüntüye dönüştürüldükten sonra (Resim 1), bu görüntüye karşılık gelen DSİ haritası üzerinde seçilen test alanlarının yüzölçümleri belirlenerek , görüntüye ait veriler seçilen sınıf örneklerine göre, farklı sınıflandırma yöntemleriyle işlenmiş ve elde edilen sonuçların analizi yapılmıştır.

4.1. Sınıflama Öncesinde Yapılan Çalışmalar

4.1.1. Test Alanlarının ve Sınıfların Belirlenmesi

Harita üzerindeki ilk çalışmada, bölgenin daha çok tahıl, patates, bostan, narenciye, toprak1 (yeni mısır ekilmiş alanlar) ve toprak2 (çıplak arazi) olarak adlandırılan örtülerden oluştuğu saptanmıştır. Uydu görüntülerinin (3., 4. ve 5. bant) gözle yorumu sonucunda da, bu örtülerin birbirinden genellikle farklı renklerle ayrık olduğu belirlenmiş ve üzerinde çalışılacak sınıflar olarak seçilmiştir.

İkinci adımda, haritadan faydalanarak yergerçeğini belirlemek amacıyla, her sınıftan ikiyeşer tane olmak üzere test alanları seçilmiş ve önce görüntünün sadece bu alanları her yöntemle ayrı ayrı sınıflandırılmıştır. Bu sınıflama işlemleri sonucunda, sınıfları temsil eden test alanlarına ait piksel sayıları elde edilmiş ve bu sayılar, haritada bu alanların m² cinsinden verilen yüzölçümlerinin 900 'e bölünmesi ($1 \text{ piksel} = 30 \times 30 \text{ m}^2 = 900 \text{ m}^2$) sonucunda bulunan piksel sayılarıyla karşılaştırılarak her yöntemin hata oranı hesaplanmıştır. Karşılaştırmalar her sınıftan ikiyeşer örneğe ait piksel sayılarının ayrı ayrı toplamı üzerinden yapılmıştır. Test alanlarının sınıflandırılması sonucunda elde edilen piksel sayıları, haritaya göre yer gerçeğine karşılık gelen piksel sayıları ve hata oranları çizelge 1 'de verilmiştir.

Karşılaştırmalar sonucunda en az hatayı Elipsoid yönteminin, en çok hatayı da Üçgen yönteminin verdiği görülmüştür. Üzerinde çalışılan uydu verisi 1992 yılına ait olduğundan yer gerçeği çalışması yapılamamıştır. Dolayısıyla bütün görüntünün yer gerçeği, yine bütün görüntünün en az hata oranına sahip Elipsoid yöntemiyle sınıflandırılması sonucunda elde edilen değerlere dayanarak ve Elipsoid yönteminin test

alanları üzerinde yapılan sınıflandırma işlemlerinde verdiği hata oranları göz önünde tutularak hesaplanmış ve diğer yöntemlerin sonuçları bu değerlerle karşılaştırılmıştır.

Çizelge 1. Yer gerçeğinin belirlenmesinde test alanlarıyla elde edilen sonuçlar ve hata oranları (R=3).

<i>Sınıf/Metod</i>	<i>Yer Gerçeği</i>	<i>Üçgen</i>	<i>Kutu</i>	<i>Elipsoid</i>	<i>Min. Mesafe</i>	<i>Max. Benzerlik</i>	<i>En Ykn. Komşu</i>
Tahul	526	342	672	494	455	460	473
%Hata		%34.98	%27.75	%6.08	%13.49	%12.54	%10.07
Patates	258	249	381	254	243	256	250
%Hata		%3.48	%47.67	%1.55	%5.81	%0.77	%3.10
Bostan	611	570	756	601	810	595	829
%Hata		%6.71	%23.73	%1.63	%32.56	%2.61	%35.67
Narenciye	422	177	535	415	361	321	321
%Hata		%58.05	%26.77	%1.65	%14.45	%23.93	%23.93
Toprak1	326	111	293	316	322	299	313
%Hata		%65.95	%10.12	%3.06	%1.22	%8.28	%3.98
Toprak2	720	412	767	677	683	656	709
%Hata		%42.77	%6.52	%5.97	%5.13	%8.88	%1.52

4.1.2. Sınıf Örneklerinin Belirlenmesi

Üzerinde çalışılacak sınıflara karar verildikten sonra, bu sınıfları sınıflandırma işleminde temsil edecek örnekler seçilmiştir. Harita ve uydu verilerinden oluşan görüntüler birlikte incelenerek, ekrandaki görüntü üzerinde 25 tane örnek alan seçilmiştir. Bu örnek alanların sınıflara göre dağılımı ve piksel sayıları, çizelge 2 'de verilmiştir.

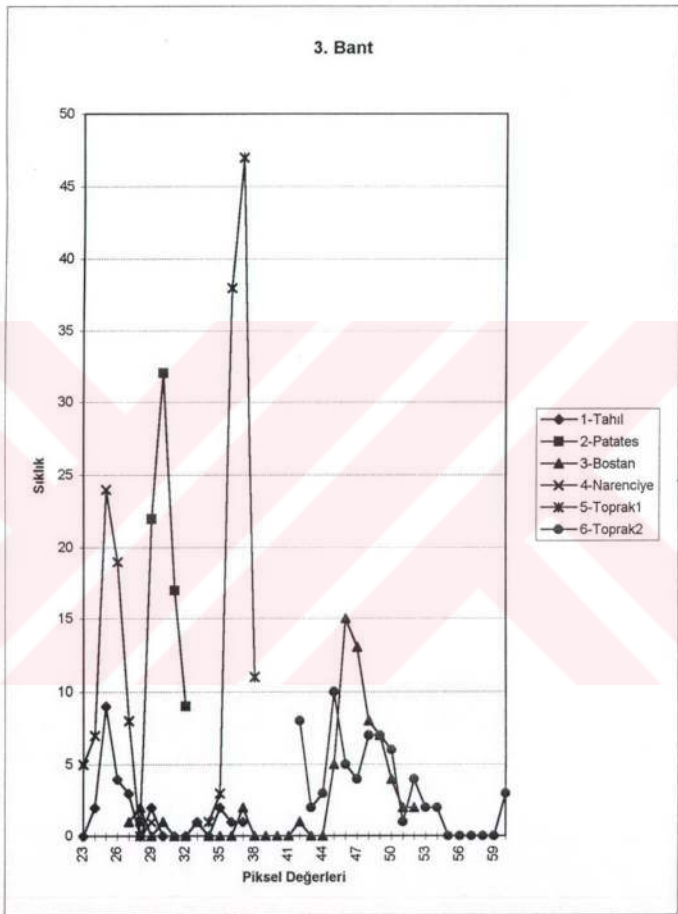
4.1.3. Sınıf Örneklerine Ait Piksellerin İstatistiksel Değerlerinin Belirlenmesi

Sınıflandırma işlemine başlamadan yapılan son işlem, seçilen 6 adet sınıfı temsil edecek 25 adet örnek alana ait piksel gruplarının ayrı ayrı ortalama piksel değerlerini ve standart sapmalarını hesaplamak olmuştur (Çizelge 2). Bu değerler incelenen bütün sınıflandırma yöntemlerinde kullanılmaktadır. Ayrıca, Maksimum Benzerlik yönteminde kullanmak üzere bu piksel gruplarının ayrı ayrı varyans-kovaryans matrisleri, kovaryans

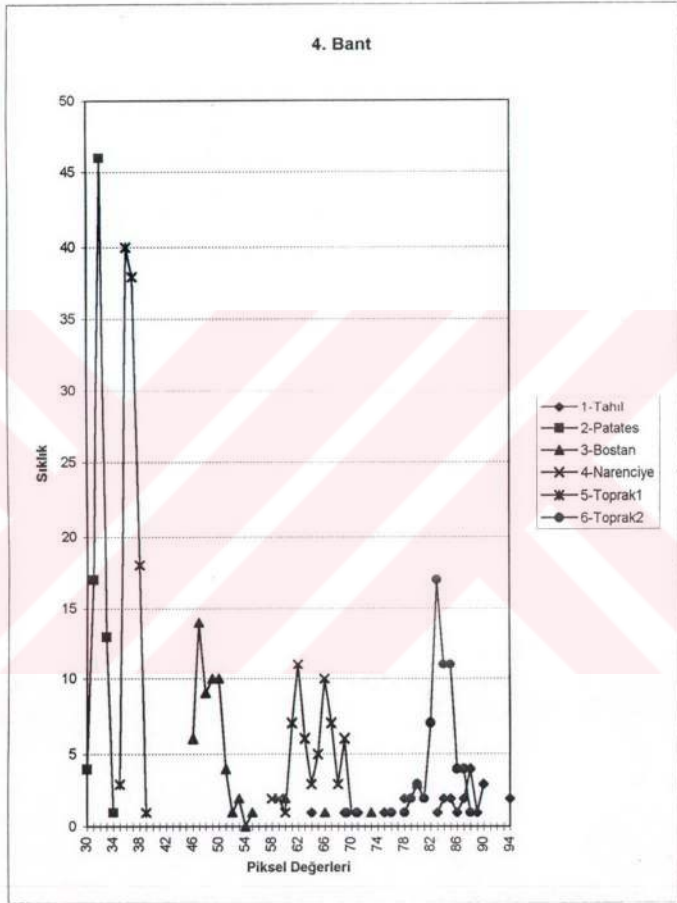
matrislerin determinantları ve kovaryans matrislerin tersleri, Üçgen yönteminde kullanmak üzere bu piksel gruplarının 3 ayrı banttaki ortalama ve standart sapma değerlerinden faydalanarak üçgen alanları hesaplanmış, En Yakın Komşu yönteminde kullanılmak üzere her pikselin parlaklık değeri hafızada saklanmıştır. Şekil 6 (a,b,c) 'da her sınıfın birinci örneğine ait 3., 4. ve 5. bantlardaki piksel değerlerinin sıklıkları, şekil 7 (a,b,c) 'de her sınıfın 3., 4. ve 5. bantlardaki ortalama piksel değerleri ve şekil 8 'de sınıfların ortalama piksel değerlerinin bantlara göre dağılımları verilmiştir.

Çizelge 2. Sınıfları temsil eden örnek alanlara ait piksellerin 3., 4., 5. bantlardaki ortalama ve standart sapma değerleri.

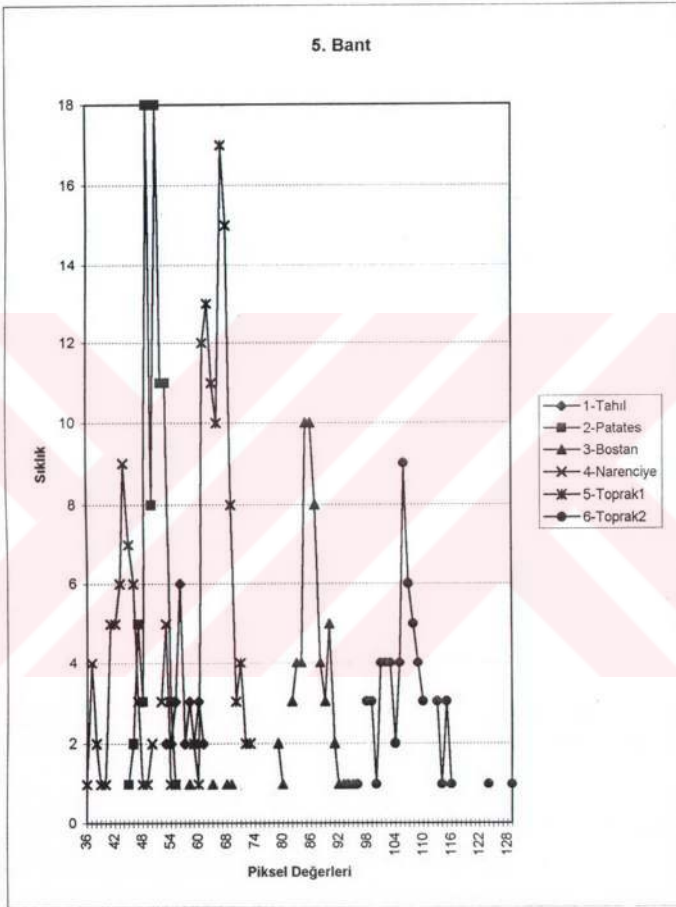
Sınıf No	Cinsi	Piksel Sayısı	3. Bant		4. Bant		5. Bant	
			Ortalama	Std. Sapma	Ortalama	Std. Sapma	Ortalama	Std. Sapma
1.1	Tahıl	25	27.68	4.08	83.64	7.72	57.08	2.72
1.2	"	81	23.09	1.42	82.49	5.14	47.83	1.99
1.3	"	25	30.72	1.06	75.28	1.06	68.92	1.82
1.4	"	49	33.42	2.87	60.94	5.75	63.47	2.55
1.5	"	81	25.35	1.20	73.64	9.61	52.43	1.46
1.6	"	100	23.91	2.51	80.21	7.68	48.01	4.90
2.1	Patates	81	30.14	0.98	31.88	0.78	50.47	2.09
2.2	"	49	34.00	1.19	33.55	0.79	52.88	1.83
2.3	"	36	33.58	1.02	33.94	0.53	59.19	0.75
2.4	"	49	34.27	0.84	35.99	3.33	59.18	2.32
3.1	Bostan	64	45.64	5.50	51.08	7.72	84.98	6.41
3.2	"	100	41.32	1.02	41.47	1.27	68.64	2.13
3.3	"	81	49.68	1.86	51.60	2.13	99.16	5.47
4.1	Narenciye	64	25.34	1.17	64.30	3.05	44.81	4.74
4.2	"	121	30.93	1.84	50.79	3.67	54.83	4.23
5.1	Toprak1	100	36.64	0.76	36.74	0.82	64.42	2.63
5.2	"	49	35.63	1.03	35.78	0.62	60.18	2.50
5.3	"	49	36.24	1.18	39.63	2.67	60.67	2.65
5.4	"	100	35.75	0.95	36.13	0.94	64.40	2.89
5.5	"	25	41.28	7.93	53.44	14.45	75.60	6.01
5.6	"	25	37.16	1.93	39.12	2.45	69.00	3.96
6.1	Toprak2	64	47.72	4.28	83.41	2.24	106.28	6.10
6.2	"	49	57.39	2.12	59.39	1.79	118.16	4.08
6.3	"	121	57.55	2.41	58.83	2.39	116.64	5.33
6.4	"	64	55.66	2.47	58.06	2.36	114.88	3.86



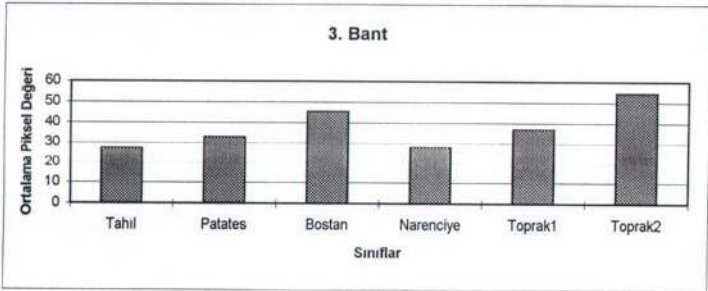
Şekil 6-a. Seçilen sınıf örneklerine ait piksellerin 3. banttaki dağılımı.



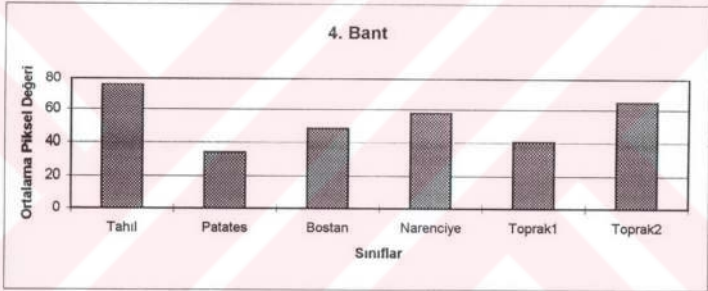
Şekil 6-b. Seçilen sınıf örneklerine ait piksellerin 4. banttaki dağılımı.



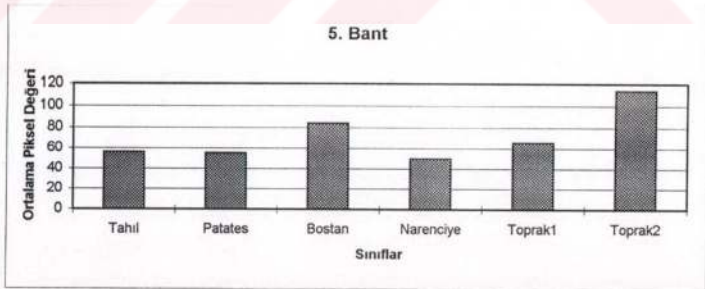
Şekil 6-c. Seçilen sınıf örneklerine ait piksellerin 5. banttaki dağılımı.



(a)

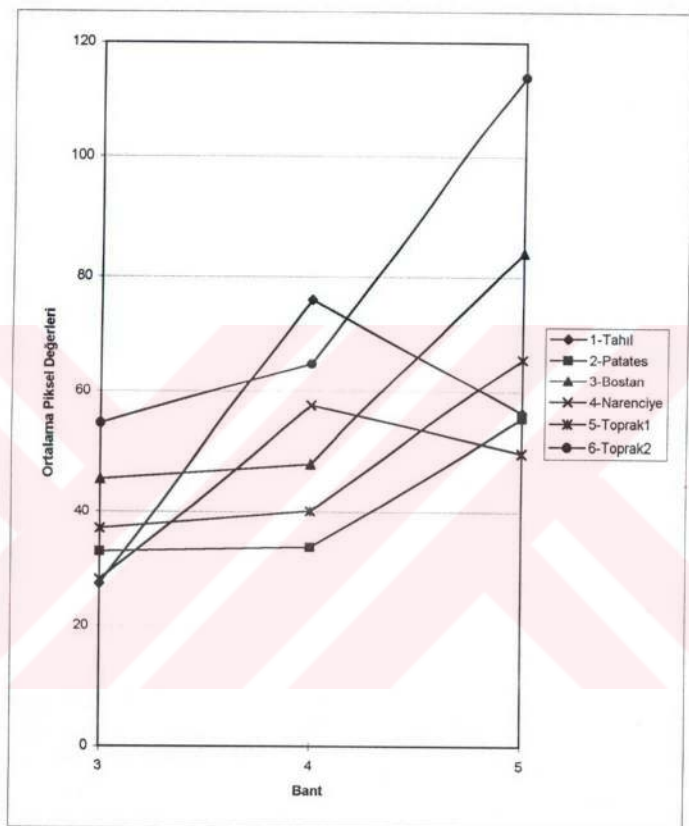


(b)



(c)

Şekil 7. Seçilen sınıf örneklerinin a) 3. bantta, b) 4. bantta, c) 5. bantta ortalama piksel değerleri.



Şekil 8. Seçilen sınıf örneklerine ait piksellerin ortalamalarının bantlara göre dağılımı.

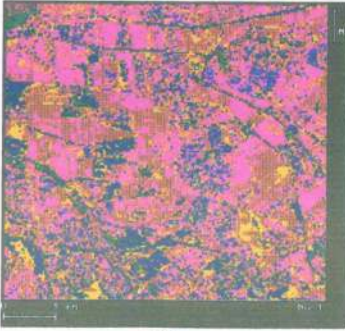
4.2. Sınıflandırma İşlemleri

Landsat TM uydusunun 3., 4. ve 5. bantlardaki veri setinin 198x200 piksellik bölümü, seçilen sınıf örneklerine ait pikseller ve bu piksellerin istatistiksel değerleri yardımıyla, hazırlanan programda (Ek-1) uygulanabilen Maksimum Benzerlik, En Yakın Komşu, Minimum Mesafe, Kutu, Elipsoid ve Üçgen yöntemleriyle ayrı ayrı sınıflandırılmış ve bu işlemler sonucunda aşağıda çizelge 3 'de verilmiş olan her bir yöntemin her bir sınıf için ürettiği piksel sayısı ve bütün görüntüyü sınıflandırma süreleri elde edilmiştir. Bazı yöntemlere ait işlemlerde kullanılan güven aralığı, $R=3$ olarak seçilmiştir. Çizelge 3 'ün en alt satırında verilmiş olan sınıfların yer gerçeğine ait piksel sayıları, tezin 4.1.1. bölümünde anlatıldığı gibi Elipsoid yönteminin ürettiği sonuçlara dayanarak hesaplanmıştır. Bu sınıflara ait yeryüzü örtülerinin m^2 cinsinden alanlarını bulabilmek için, her birini ayrı ayrı 900 ile çarpmak gerekmektedir.

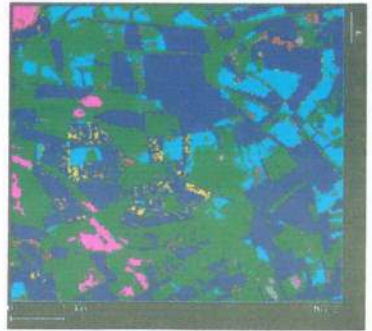
Çizelge 3. Sınıflandırma sonuçları ($R=3$).

Yöntem/Sınıf	Tahıl	Patates	Bostan	Narenci	Tprk1	Tprk2	Toplam	Süre(s)
Üçgen	7,163	145	891	7,567	8,440	14,715	38,921	123
Kutu	15,381	5,472	15,468	724	921	1,313	39,279	128
Elipsoid	13,349	3,640	11,274	1,380	7,111	1,460	38,214	486
Min. Mesafe	13,236	2,649	8,953	2,964	9,767	2,031	39,600	527
Max. Benz.	11,116	2,433	8,801	3,761	11,314	2,175	39,600	1,414
En Ykn Kmş	13,951	2,950	4,985	2,926	12,472	2,316	39,600	26,654
Yer Gerçeği	14,160	3,696	11,457	1,403	7,337	1,547	39,600	

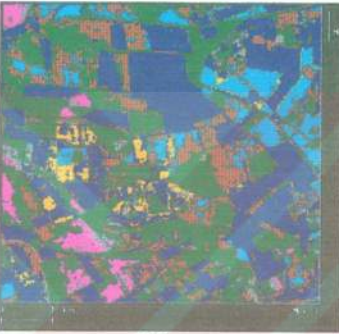
Sınıflandırma işlemleri sonucunda her bir yöntemin ürettiği sınıflanmış görüntü ve bu görüntülerdeki renklerin hangi sınıfları temsil ettiği, resim 2 (a, b, c, d, e, f) 'de verilmiştir. Bu resimler, üzerinde olduğu bilgisayar ekranından olduğu gibi aktarılmıştır.



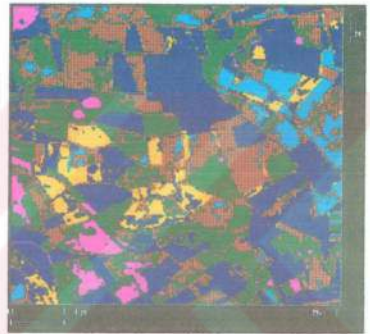
(a)



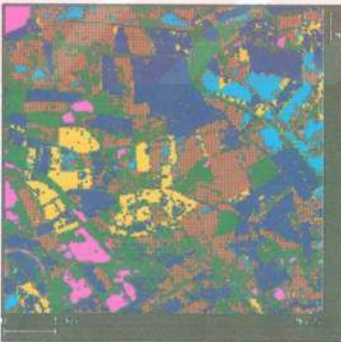
(b)



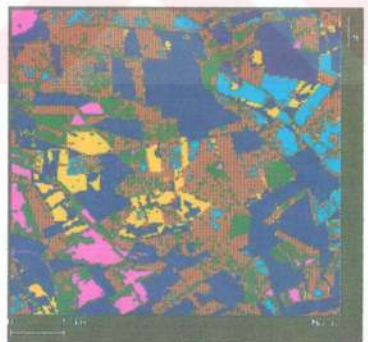
(c)



(d)



(e)



(f)

Resim 2. Sınıflanmış görüntüler. a) Üçgen yöntemi, b) Kutu yöntemi, c) Elipsoid yöntemi, d) Min. Mesafe yöntemi, e) Max. Benzerlik yöntemi, f) En Ykn Komşu yöntemi.

■ -Tahıl, ■ -Patates, ■ -Bostan, ■ -Narenciye, ■ -Toprak1, ■ -Toprak2.

4.2.1. Sınıflandırma Yöntemlerinin İşlem Hızlarının Karşılaştırılması

Landsat TM veri setinden oluşturulan 198x200 piksellik görüntünün, 6 adet yöntemle ayrı ayrı sınıflandırılması sonucunda her bir yöntemin uygulanma süresi (bilgisayar zamanı) belirlenmiş ve aşağıdaki çizelge 4 'te verilmiştir. Bu çizelgeden de görüldüğü gibi en hızlı Üçgen yöntemi olmak üzere (123 saniye), ilk 5 yöntem sınıflandırma işlemini çok kısa sürelerde yapmasına rağmen, son sırada olan En Yakın Komşu yöntemi aynı işlemi olağanüstü uzun bir sürede (7 saat, 24 dakikada) gerçekleştirmiştir.

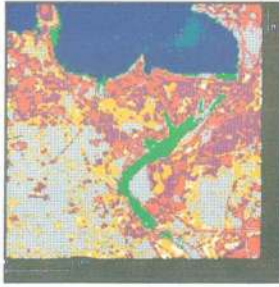
Çizelge 4. Sınıflandırma yöntemlerinin işlem süreleri.

	Sınıflandırma Yöntemleri					
	Üçgen	Kutu	Elipsoid	Minimum Mesafe	Maksimum Benzerlik	En Yakın Komşu
Süre (saniye)	123	128	486	527	1,414	26,654

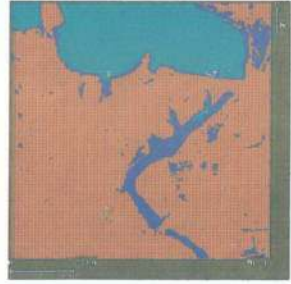
4.2.2. Yeni Algoritmanın (Üçgen) Başka Bir Veri Seti Üzerinde Denenmesi

Yeni düşünülen algoritmanın, Landsat-TM veri setini sınıflandırması işleminde çok büyük hatalar vermesinden dolayı, bu algoritma başka bir veri setinin (SPOT, Adana ili) 200x200 piksellik bir bölümü üzerinde (Resim 3-a), sadece su ve karayı ayırmak amacıyla denenmiştir. Bu veri setinden oluşturulan görüntü üzerinde, karar verilen 3 adet sınıfı (göl, nehir, kara) temsil eden 24 adet örnek alan seçilmiş ve bu görüntü, Landsat-TM veri seti için hazırlanan aynı programla (Ek-1) sınıflandırılmıştır.

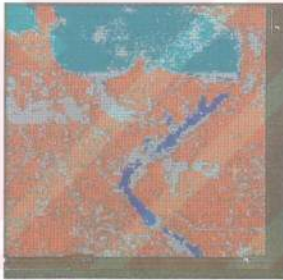
Bu algoritmayla birlikte aynı veri seti, karşılaştırma yapmak amacıyla Elipsoid ve Maksimum Benzerlik yöntemleriyle de sınıflandırılmış, bu işlemler sonucunda elde edilen görüntüler, SPOT veri setinden oluşturulan görüntüyle birlikte resim 3 'te verilmiştir. Resim 3-b 'de görüldüğü gibi Üçgen yöntemi, su ve karayı ayırmada daha iyi bir sonuç vermiştir.



a)



b)



c)



d)

Resim 3. SPOT uydusuna ait veri setinin ham ve sınıflanmış görüntüleri. a) SPOT verisi, b) Üçgen yöntemi, c) Elipsoid yöntemi d) Maksimum Benzerlik yöntemi.

■ - Göl ■ - Nehir ■ - Kara ■ - Sınıflanmayan

5. TARTIŞMA ve SONUÇLAR

Çalışmada uygulanan Üçgen, Kutu, Elipsoid, Minimum Mesafe, Maksimum Benzerlik ve En Yakın Komşu yöntemlerinin birbirlerine göre doğruluğunu araştırmak için bu yöntemler arasındaki ilişkileri verebilecek, sınıflandırmada kayıp piksel toplam yüzdesi (KPTY),

$$KPTY = \sum_i |P_i - \hat{P}_i| \quad (5.1)$$

ve kayıp piksel olasılığının normalize edilmiş toplamı (KPONT),

$$KPONT = \frac{\left(\sum_i |P_i - \hat{P}_i| \cdot P_i \right)}{P_{\text{Üçgen}}} \quad (5.2)$$

bağıntılarıyla hesaplanmıştır. Burada,

M_i : Yer gerçeğine göre i . sınıfın piksel sayısı,

$\hat{M}_i$: Sınıflandırma sonucunda bulunan i . sınıfın piksel sayısı,

olmak üzere,

$$M = \sum_i M_i, \quad P_i = \frac{M_i}{M} \quad \text{ve} \quad \hat{P}_i = \frac{\hat{M}_i}{M} \quad \text{dir.}$$

Eşitlik 5.2 'deki normalizasyon işlemi $(P_{\text{Üçgen}} = \sum_i |P_i - \hat{P}_i| \cdot P_i)$ Üçgen yöntemine göre yapılmıştır.

Sınıflandırılan alanda bulunan tahıl, patates, bostan, narenciye, toprak1 ve toprak2 olarak adlandırılan örtü tiplerinin, yer gerçeği ve sınıflandırma sonuçlarına göre elde edilen piksel sayıları (M_i ve $\hat{M}_i$) çizelge 3 'te verilmişti. Bu değerlerden hesaplanan sınıfların kapladıkları alan yüzdeleri çizelge 5 'te, KPTY ve KPONT değerleri ise Üçgen yöntemi için çizelge 6 'da, Kutu yöntemi için çizelge 7 'de, Elipsoid yöntemi için çizelge 8 'de, Minimum Mesafe yöntemi için çizelge 9 'da, Maksimum Benzerlik yöntemi için çizelge 10 'da ve En Yakın Komşu yöntemi için ise çizelge 11 'de verilmiştir.

Çizelge 5. Eğitimli sınıflandırma sonucunda elde edilen sınıfların kapladıkları alan yüzdeleri.

<i>Yöntem</i>	<i>Tahıl</i> %	<i>Patates</i> %	<i>Bostan</i> %	<i>Narenci</i> %	<i>Toprak1</i> %	<i>Toprak2</i> %	<i>Sınıflan- mayan %</i>
<i>Üçgen</i>	18.09	0.37	2.25	19.11	21.31	37.16	1.71
<i>Kutu</i>	38.84	13.82	39.06	1.83	2.32	3.31	0.82
<i>Elipsoid</i>	33.71	9.19	28.47	3.48	17.96	3.69	3.50
<i>Min. Mesafe</i>	33.42	6.69	22.61	7.48	24.66	5.14	0.00
<i>Max. Benz.</i>	28.07	6.14	22.22	9.50	28.57	5.50	0.00
<i>En Yk Komşu</i>	35.23	7.45	12.59	7.39	31.49	5.85	0.00
<i>Yer gerçeği</i>	35.76	9.33	28.93	3.54	18.53	3.91	0.00

Çizelge 6. Üçgen yöntemi ile her bir örtü tipi için sınıflanamayan piksel yüzdeleri ve sınıflanamayan piksel olasılıkları.

<i>Sınıf</i>	$P_i = \frac{M_i}{M}$	$\hat{P}_i = \frac{\hat{M}_i}{M}$	$ P_i - \hat{P}_i $	$ P_i - \hat{P}_i \cdot P_i$
<i>Tahıl</i>	0.3576	0.1809	0.1767	0.063180
<i>Patates</i>	0.0933	0.0037	0.0896	0.008360
<i>Bostan</i>	0.2893	0.0225	0.2668	0.077190
<i>Narenciye</i>	0.0354	0.1911	0.1557	0.005512
<i>Toprak1</i>	0.1853	0.2131	0.0278	0.005151
<i>Toprak2</i>	0.0391	0.3716	0.3325	0.013000
<i>Toplam</i>			1.0491	0.172393

Çizelge 7. Kutu yöntemi ile her bir örtü tipi için sınıflanamayan piksel yüzdeleri ve sınıflanamayan piksel olasılıkları.

<i>Sınıf</i>	$P_i = \frac{M_i}{M}$	$\hat{P}_i = \frac{\hat{M}_i}{M}$	$ P_i - \hat{P}_i $	$ P_i - \hat{P}_i \cdot P_i$
<i>Tahıl</i>	0.3576	0.3884	0.0308	0.011010
<i>Patates</i>	0.0933	0.1382	0.0449	0.004420
<i>Bostan</i>	0.2893	0.3906	0.1013	0.029310
<i>Narenciye</i>	0.0354	0.0183	0.0171	0.000610
<i>Toprak1</i>	0.1853	0.0232	0.1621	0.030030
<i>Toprak2</i>	0.0391	0.0332	0.0059	0.000230
<i>Toplam</i>			0.3621	0.075610

Çizelge 8. Elipsoid yöntemi ile her bir örtü tipi için sınıflanamayan piksel yüzdeleri ve sınıflanamayan piksel olasılıkları.

<i>Sınıf</i>	$P_i = \frac{M_i}{M}$	$\hat{P}_i = \frac{\hat{M}_i}{M}$	$ P_i - \hat{P}_i $	$ P_i - \hat{P}_i \cdot P_i$
<i>Tahıl</i>	0.3576	0.3371	0.0205	0.007331
<i>Patates</i>	0.0933	0.0919	0.0014	0.000131
<i>Bostan</i>	0.2893	0.2847	0.0046	0.001331
<i>Narenciye</i>	0.0354	0.0348	0.0006	0.000021
<i>Toprak1</i>	0.1853	0.1796	0.0057	0.001056
<i>Toprak2</i>	0.0391	0.0369	0.0022	0.000086
<i>Toplam</i>			0.0350	0.009956

Çizelge 9. Minimum Mesafe yöntemi ile her bir örtü tipi için sınıflanamayan piksel yüzdeleri ve sınıflanamayan piksel olasılıkları.

<i>Sınıf</i>	$P_i = \frac{M_i}{M}$	$\hat{P}_i = \frac{\hat{M}_i}{M}$	$ P_i - \hat{P}_i $	$ P_i - \hat{P}_i \cdot P_i$
Tahıl	0.3576	0.3342	0.0234	0.008370
Patates	0.0933	0.0669	0.0264	0.002460
Bostan	0.2893	0.2261	0.0632	0.018280
Narenciye	0.0354	0.0748	0.0394	0.001390
Toprak1	0.1853	0.2466	0.0613	0.011360
Toprak2	0.0391	0.0514	0.0123	0.000480
Toplam			0.2260	0.042340

Çizelge 10. Maksimum Benzerlik yöntemi ile her bir örtü tipi için sınıflanamayan piksel yüzdeleri ve sınıflanamayan piksel olasılıkları.

<i>Sınıf</i>	$P_i = \frac{M_i}{M}$	$\hat{P}_i = \frac{\hat{M}_i}{M}$	$ P_i - \hat{P}_i $	$ P_i - \hat{P}_i \cdot P_i$
Tahıl	0.3576	0.2807	0.0769	0.027490
Patates	0.0933	0.0614	0.0319	0.002970
Bostan	0.2893	0.2222	0.0671	0.019410
Narenciye	0.0354	0.0950	0.0596	0.002110
Toprak1	0.1853	0.2857	0.1004	0.018600
Toprak2	0.0391	0.0550	0.0159	0.000620
Toplam			0.3518	0.071200

Çizelge 11. En Yakın Komşu yöntemi ile her bir örtü tipi için sınıflanamayan piksel yüzdeleri ve sınıflanamayan piksel olasılıkları.

Sınıf	$P_i = \frac{M_i}{M}$	$\hat{P}_i = \frac{\hat{M}_i}{M}$	$ P_i - \hat{P}_i $	$ P_i - \hat{P}_i \cdot P_i$
Tahıl	0.3576	0.3523	0.0053	0.001895
Patates	0.0933	0.0745	0.0188	0.001754
Bostan	0.2893	0.1259	0.1634	0.047271
Narenciye	0.0354	0.0739	0.0385	0.001363
Toprak1	0.1853	0.3149	0.1296	0.024014
Toprak2	0.0391	0.0585	0.0194	0.000759
Toplam			0.3750	0.077056

Çizelge 12. Eğitilmiş sınıflandırma yöntemlerinin sınıflanamayan toplam piksel yüzdesi ve normalize edilmiş sınıflanamayan toplam piksel olasılığına göre birbirleriyle ilişkileri.

Yöntem	Sınıflanamayan Toplam Piksel Yüzdesi $KPTY = \sum_i P_i - \hat{P}_i \times 100$	Normalize Edilmiş Sınıflanamayan Toplam Piksel Olasılığı $KPONT = \left(\sum_i P_i - \hat{P}_i \cdot \hat{P}_i \right) / P_{Oggen}$
Üçgen	104.91	1
Kutu	36.21	0.439
Elipsoid	3.50	0.058
Minimum Mesafe	22.60	0.246
Maksimum Benzerlik	35.18	0.413
En Yakın Komşu	37.50	0.447

Tüm alanlar için KPONT değerleri Üçgen yöntemine göre normalize edilmiş ve karşılaştırmalar bu değere göre yapılmıştır. Çizelge 12 'de, uygulanan tüm yöntemlerin KPTY ve KPONT değerleri görülmektedir. Sınıflandırma sonuçlarına göre Üçgen yöntemi çok hatalı sonuçlar verirken, Elipsoid yöntemi diğer yöntemlere göre en iyi sonuçları vermiştir. Elipsoid yöntemi için hesaplanan KPTY değeri %3.50 ve KPONT değeri ise 0.058 dir. Bunun yanında çizelge 12 'den de görüleceği gibi Elipsoid yöntemine en yakın değerleri veren Minimum Mesafe yöntemi için KPTY değeri %22.60 ve KPONT değeri ise 0.246 olmuştur. Bu değerler Elipsoid yöntemi için elde edilen değerlerin oldukça üzerindedir. Diğer yöntemler için elde edilen değerler çizelge 12 'de görüldüğü gibi daha da yüksektir.

Çalışmada kullanılan tüm eğitilmiş sınıflama yöntemlerinin bilgisayar zamanları hesaplanmış ve bu değerler çizelge 13 'te verilmiştir. Süreler yukarıda olduğu gibi Üçgen yöntemi için hesaplanan bilgisayar zamanı ile normalize edilmiştir. Yöntemlerin toplam performansları (TP),

$$TP = \left[\frac{1}{KPONT \times \% Bilgisayar Zamanı} \right]^{1/2} \quad (5.3)$$

bağıntısı kullanılarak hesaplanmış ve elde edilen sonuçlar aşağıda çizelge 13 'te verilmiştir.

Çizelge 13. Eğitilmiş sınıflandırma yöntemlerinin normalize edilmiş bilgisayar zamanlarına göre işlem hızları ve toplam performansları.

Yöntem	Süre (Saniye)	Normalize Edilmiş % Bilgisayar Zamanı	Normalize Edilmiş Toplam Performans
Üçgen	123	1	1
Kutu	128	1.04	1.48
Elipsoid	486	3.95	2.09
Minimum Mesafe	527	4.28	0.98
Maksimum Benzerlik	1,414	11.50	0.46
En Yakın Komşu	26,654	216.70	0.10

Bu çizelgeden de görüldüğü gibi Üçgen yönteminin toplam performansının, Minimum Mesafe, Maksimum Benzerlik ve En Yakın Komşu yöntemine göre daha yüksek çıkması bilgisayar zamanının çok kısa olmasından, bunun yanında anılan diğer yöntemlerin bilgisayar zamanlarının çok uzun olmasındandır. Normalize edilmiş toplam performans değerlerine bakıldığında Elipsoid yönteminin en iyi olduğu görülmektedir.

Yapılan sınıflandırma çalışmalarında elde edilen görüntüler (Resim2 ve Resim 3) ve yukarıdaki çizelgelerde sıralanmış değerler, Üçgen yönteminin Landsat-TM veri setine ait görüntüde olduğu gibi piksel parlaklık değerleri birbirlerine çok yakın olan örtü tiplerini doğru olarak sınıflandıramadığını, fakat SPOT verisi üzerindeki denemede olduğu gibi su ve kara gibi piksel parlaklık değerleri birbirinden çok farklı olan örtü tiplerini daha iyi sınıflandırabildiğini göstermektedir.

ÖZET

Bu çalışmada, uzaktan algılama tekniği ile elde edilen verileri sınıflandırmak amacıyla geliştirilmiş eğitimli sınıflandırma yöntemleri (Kutu, Elipsoid, Minimum Mesafe, Maksimum Benzerlik ve En Yakın Komşu) incelenmiş ve aralarındaki performans ilişkileri belirlenerek, doğruluk ve işlem hızları açısından birbirleriyle karşılaştırılmıştır. Bunlarla birlikte, aynı amaçlı yeni bir algoritma (Üçgen) denenmiştir. Bu yöntemler, Adana ilindeki tarımsal bir bölgeye ait 27.03.1992 tarih ve koordinatı 175-34 olan LANDSAT-TM veri setinin 198x200 piksellik bölümü ile Adana ili merkezine ait SPOT veri setinin 200x200 piksellik bölümü üzerinde, hazırlanan bir görüntü işleme programıyla ayrı ayrı uygulanmıştır.

Sınıflandırma işlemleri yapılmadan önce, uydu verileri görüntüye dönüştürülmüş ve bu görüntüye karşılık gelen DSİ haritası üzerinde yapılan ilk çalışmada, bölgenin daha çok tahıl, patates, bostan, narenciye, toprak1 (yeni mısır ekilmiş alanlar) ve toprak2 (çıplak arazi) olarak adlandırılan örtülerden oluştuğu saptanmıştır. Seçilen bu sınıflara ait yer gerçeğini belirlemek amacıyla, önce test alanlarıyla denemeler yapılmış ve her yöntemin hata oranı saptanmıştır. Daha sonra, haritadan faydalanarak sınıflanacak görüntü üzerinde sınıflandırma işleminde sınıfları temsil edecek örnek alanlar seçilmiş ve bu örnek alanlara ait piksellerin parlaklık değerlerinden, yöntemler için gerekli olan ortalama, varyans, standart sapma ve varyans-kovaryans matrisleri gibi istatistiksel değerler hesaplanmıştır.

Bu aşamalardan sonra, Landsat-TM görüntüsü tüm yöntemlerle ayrı ayrı sınıflandırılarak oluşan sınıflara ait piksel sayıları ve yöntemlerin bilgisayar süreleri değerlendirilerek performans ilişkileri belirlenmiştir. Belirlenen bu sonuçlar, Elipsoid yönteminin en fazla performans ve doğruluğa sahip olduğunu, En Yakın Komşu yönteminin uygulama süresinin aşırı bir şekilde fazla olması nedeniyle en düşük performansa sahip olduğunu ve Üçgen yönteminin uygulama süresinin en az olduğunu fakat çok hatalı sonuçlar verdiğini göstermiştir. Ancak, Üçgen yönteminin SPOT verisi üzerinde su ve kara gibi iki ayrı örtü tipini sınıflandırmak için yapılan ikinci bir denemede daha iyi sonuçlar alınmıştır.

SUMMARY

In this study, the supervised classification methods (Box, Elipsoid, Minimum Distance, Maximum Likelihood and Nearest Neighbor) were compared each other by applying to the remote sensing data according to the accuracy and computer processing time. However, a new algorithm (Triangle method) was also examined. These methods have been applicated on a part of 198x200 pixels of LANDSAT-TM image of an agricultural area and a part of 200x200 pixels of SPOT image of centrum in Adana City by home made programs about image processing.

Before classification, different cover samples have been selected on the image by comparing with DSI map and classification have been made. After the ground truth, samples of areas belong the covers to represent the known classes have been selected to use for classification processings and computed the values of the statistics (e.g. mean, standard deviation, variance-covariance matrices) from pixel values of this samples.

Later these steps, the image of Landsat-TM have been classified one by one with every methods. The numbers of pixels of occurent classes and computer processing time have been determined for each method, and carried out the relatives of performances among these methods. Results of this study have been showed that Elipsoid method had got the best performance and accuracy, the application time of Nearest Neighbour method was too much and Triangle method had got the worst accuracy. But, second application have been made on the SPOT data by Triangle method to separate water and land. And better results have been determined by this method.

KAYNAKLAR

- ARAI, K., 1992. A supervised thematic mapper classification with a purification of training samples. *Int. J. Remote Sensing*, Vol. 13, No. 11, 2039-2049.
- CONESE, C., MARACCHI, G. and MASELLI, F., 1993. Improvement in Maximum Likelihood Classification performance on highly rugged terrain using Principal Components Analysis. *Int. J. Remote Sensing*, Vol. 14, No. 7, 1371-1382.
- DİNÇ, U., YEĞİNGİL, İ., İNCE, F., ŞENOL, S., PEŞTEMALCI, V. ve ERTUNG, Ö., 1986. Uzaktan Algılama Yaz Okulu Ders Kitabı. Ç.Ü. Fen Bilimleri Enstitüsü Yayınları. Adana.
- EMRAHOĞLU, N., 1993. Eğitilmiş Sınıflama Metodlarının Karşılaştırması ve Yeni Algoritmanın Denenmesi. Ç.Ü. Fen Bil. Ens., Fizik Anabilim Dalı, Doktora Tezi, Adana.
- FITZGERALD, R.W. and LEES, B.G., 1994. Assessing the classification accuracy of multisource remote sensing data. *Remote Sensing of Environment*, Vol. 47, No. 3, 362-368.
- FRANKLIN, S.E. and WILSON, B.A., 1991. Spatial and spectral classification of remote sensing imagery. *Computers&Geosciences*, Vol. 17, No. 8, 1151-1172.
- HABERACKER, P., 1984. Digital Image Processing. Workshop on Remote Sensing Applications in Civil Engineering, University of Dundee, Scotland, UK.
- İNCE, F., 1986. Maximum Likelihood Classification, Optimal or Problematic?. A Comparison with the KNN Classification. Tübitak MAM Elektronik Araştırma Bölümü. Technical Report No: TR-86/09.
- INOMATA, Y. and OGATA, S. 1993. Supervised and unsupervised classification by histogram overlay techniques. *Int. J. Remote Sensing*, Vol. 14, No. 14, 2605-2616.

- KARTIKEYAN, B., GOPALAKRISHNA, B., KALUBARME, M.H. and MAJUMDER, K.L., 1994. Contextual techniques for classification of high and low resolution remote sensing data. *Int. J. Remote Sensing*, Vol. 15, No. 5, 1037-1051.
- MITİŞ, B., 1992. Unsupervised Sınıflandırma Yöntemlerinin İncelenmesi, Karşılaştırılması ve Yeni Metod Geliştirilmesi Üzerinde Çalışmalar. Ç.Ü. Fen Bil. Ens., Toprak Anabilim Dalı, Yüksek Lisans Tezi, Adana.
- NIBLACK, W., 1986. *An Introduction to Digital Image Processing*. Prentice-Hall International. London. (225) S.
- PEŞTEMALCI, V., DİNÇ, U., YEĞİNGİL, İ., KANDIRMAZ, M., ÇULLU, M.A., ÖZTÜRK, N. and AKSOY, E., 1995. Acreage estimation of wheat and barley fields in the province of Adana, Turkey. *Int. J. Remote Sensing*, Vol. 16, No. 6, 1075-1085.
- SHIH, F.Y. and CHEN, G.P., 1994. Classification of Landsat remote sensing images by a Fuzzy unsupervised clustering algorithm. *Information Sciences*, Vol. 1, No. 2, 97-116.
- TAKAGI, N. and KENMOCHI, M., 1991. A study of the accuracy of land coverage classification with remote sensing data. *Energy and Buildings*, Vol. 16, No. 3, 1063-1066.
- WANG, F., 1990. Fuzzy supervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 28, No.2, 194-201.
- WANG, M. and HOWARTH, P.J., 1993. Modeling errors in remote sensing image classification. *Remote Sensing of Environment*, Vol. 45, No. 3, 261-271.
- YEĞİNGİL, İ., 1987. Görüntü işleme metodları ve kümeleme tekniklerinin uygulanması. *E.Ü. Bilgisayar Araştırma ve Uygulama Merkezi Dergisi*, Cilt. 10, Sayı. 2, 193-207.

TEŞEKKÜR

Bu çalışmayı öneren ve yönlendiren, sürekli olarak ilgi ve yardımlarını gördüğüm değerli hocam Yrd. Doç. Dr. Vedat PEŞTEMALCI 'ya sonsuz saygı ve teşekkürlerimi sunarım.

Yüksek lisans öğrenimim ve tez çalışmalarım boyunca, zaman ve desteğini esirgemeyen değerli hocam Prof. Dr. İlhami YEĞİNGİL 'e sonsuz saygı ve teşekkürlerimi sunarım.

Daha önce görev yaptığım M.K.Ü. Antakya Meslek Yüksekokulu müdürleri Sayın Prof. Dr. Ali BAŞÇETİNÇELİK 'e, Sayın Prof. Dr. Cemal YÜKSELEN 'e, Sayın Prof. Dr. Mehmet AYDIN 'a ve şu anda görev yapmakta olduğum ME.Ü. Mühendislik Fakültesi Dekanı Sayın Prof. Dr. Yusuf ZEREN 'e bu çalışmayı sürdürebilmem için verdikleri izin ve teşvikleri nedeniyle sonsuz saygı ve teşekkürlerimi sunarım.

Aynı konuda çalışmalarını sürdüren, bilgi ve kaynaklarıyla çalışmamı destekleyen değerli hocam Dr. Nuri EMRAHOĞLU 'na, istatistik konusunda katkıda bulunan değerli arkadaşım Saygun KARBEYAZ'a, fotoğrafları çeken değerli arkadaşım Ahmet ÖZBEK'e, tezin basımında yardımcı olan değerli arkadaşım Cengiz YALÇIN 'a ve değerli bacanağım Levent BATU 'ya sonsuz saygı ve teşekkürlerimi sunarım.

Antakya 'da görev yapmaktayken, teşviklerinin yanı sıra çok yakın ilgi ve dostluklarını gördüğüm, çok özlediğim arkadaşlarıma ve burada adlarını sayamadığımdan dolayı çok üzüldüğüm, benim üzerimde emeği olan bütün hocalarıma ve dostlarıma sonsuz saygı ve teşekkürlerimi sunarım.

Yüksek lisans çalışmalarım süresince gösterdiği anlayış ve fedakarlığından dolayı, sevgili eşim Nuray SEÇİLMİŞ 'e sonsuz saygı ve teşekkürlerimi sunarım.

ÖZGEÇMİŞ

1964 yılında Adıyaman'da doğdum. İlk, orta ve lise öğrenimimi Adana'da tamamladım. 1986 yılında Ç.Ü. Fen-Edb. Fakültesi Fizik Mühendisliği Bölümünden mezun oldum. Askerlik görevimi tamamladıktan sonra, 1989 ve 1995 yılları arasında Hatay Mustafa Kemal Üniversitesi Antakya Meslek Yüksekokulunda öğretim görevliliği yaptım. 1992-93 döneminde Ç.Ü. Fen Bilimleri Enstitüsü Fizik Anabilim Dalında yüksek lisans programına başladım. Şu anda Mersin Üniversitesi Mühendislik Fakültesinde mühendis olarak görev yapmaktayım. Evli ve bir çocuk babasıyım.

EK-1

' Bu program, görüntü işleme ve sınıflandırma yöntemlerinin uygulanması amacıyla hazırlanmıştır.

' *Ana Prosedür* Bu bölümde Alt Prosedürlerin bildirimi ve bütün alt prosedürler tarafından paylaşılabacak değişkenlerin tanımlanması yapılmaktadır.

```

DECLARE SUB DosyayaYaz (Yontem$)
DECLARE SUB GorZoom ()
DECLARE SUB SinifHistog ()
DECLARE SUB Renk255 ()
DECLARE FUNCTION SaatFarki! (Saat1$, Saat2$)
DECLARE SUB EnYakinKomsu ()
DECLARE SUB MaxBenzerlik ()
DECLARE SUB Kovaryans ()
DECLARE SUB EnKucukMes ()
DECLARE SUB Elipsoid ()
DECLARE SUB SinifNoHes ()
DECLARE SUB KutuMetodu ()
DECLARE SUB Sinifla ()
DECLARE SUB UcgenMetodu ()
DECLARE SUB OrnekAlan ()
DECLARE SUB Boyutlar ()
DECLARE SUB RenkBilesimi ()
DECLARE SUB YogKesimi ()
DECLARE SUB histog5 ()
DECLARE SUB histog4 ()
DECLARE SUB menu2 ()
DECLARE SUB histog3 ()
DECLARE SUB histog1 ()
DECLARE SUB histog2 ()
DECLARE SUB zoom ()
DECLARE SUB gericiz ()
DECLARE SUB kareciz ()
DECLARE SUB gezgin ()
DECLARE SUB gorciz ()
DECLARE SUB Menu1 ()
DECLARE SUB menu ()
CONST Pi = 3.14
COMMON SHARED x1, x2, y1, y2, renkO%, m, n, m1, n1, min, max, Bs%, s1
COMMON SHARED Orthis3, Orthis4, Orthis5, Ss3, Ss4, Ss5
DIM SHARED detSigma(26), TersSi(26, 3, 3), Piksel(26, 3, 225) AS INTEGER
DIM SHARED i, sec$, RR AS INTEGER, Sayi55(26) AS INTEGER
DIM SHARED a(100), b(100), c(100), d(100) AS INTEGER
DIM SHARED his(256) AS INTEGER, his3(256) AS INTEGER, Alan1(26), Alan2(26) AS INTEGER
DIM SHARED kenar(6, 26) AS INTEGER, SinifNo(26) AS INTEGER, RenkKodu(15) AS INTEGER
DIM SHARED Ort(3, 26) AS INTEGER, Ss(3, 26) AS INTEGER

```

CALL menu ' "menu" prosedürünün çağırılması.

SUB Boyutlar

' Görüntü boyutlarının, bant sayısının ve ekranın tanımlanması.

```
LOCATE 9, 40: INPUT "İLK SATIR NO : ", m
LOCATE 10, 40: INPUT "SATIR SAYISI : ", m1
LOCATE 11, 40: INPUT "İLK SÜTUN NO : ", n
LOCATE 12, 40: INPUT "SÜTUN SAYISI : ", n1
LOCATE 14, 40: INPUT "GÜVEN ARALIĞI : ", RR
IF m1 > 460 THEN m1 = 460
IF n1 > 640 THEN n1 = 640
```

DO

```
LOCATE 16, 40: INPUT "BANT SAYISI : ", Bs%
```

```
LOOP UNTIL Bs% = 1 OR Bs% = 3
```

```
CLS : SCREEN 12
```

```
VIEW (0, 0)-(638, 420), 8, 5
```

```
IF Bs% = 1 THEN
```

```
CALL gorciz
```

```
ELSEIF Bs% = 3 THEN
```

```
CALL RenkBilesimi
```

```
END IF
```

```
END SUB
```

' Seçilen bant sayısına göre

' "gorciz" prosedürünün

' veya

' "RenkBilesimi" prosedürünün

' çağırılması.

SUB DosyayaYaz (Yontem\$)

' Sınıflanmış görüntülerin bir dosyaya yazılması.

```
OPEN "R", #9, Yontem$, 200
FIELD #9, 200 AS Deh$
FOR jej = 1 TO 200
De$ = ""
FOR iei = 1 TO 200
De$ = De$ + CHR$(POINT(iei, jej) + 64)
NEXT iei
LSET Deh$ = De$
PUT #9, jej
NEXT jej
CLOSE #9
END SUB
```

SUB Elipsoid

' Elipsoid yönteminin uygulanması.

```
Saat1$ = TIMES
```

```
CLS
```

```
VIEW (0, 0)-(638, 420), 8, 5
```

```

DIM Dosy$(3), A1$(3), AH1$(3), Edge%(3), PikselE(26), rt(26), DistE(26)
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz... "; : COLOR 7
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
ElipsoidPikselTop = 0
FOR i = m TO m1 + m
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn
FOR j = n TO n1 + n
trr = 0
FOR tr = 1 TO s1
Elips = 0
FOR dn = 1 TO 3
Edge%(dn) = ASC(MID$(A1$(dn), j, 1))
Elips = Elips + ((Edge%(dn) - Ort(dn, tr)) / (RR * Ss(dn, tr))) ^ 2
NEXT dn
IF Elips <= 1 THEN
trr = trr + 1: rt(trr) = tr
DistE(trr) = Elips
END IF
NEXT tr
IF trr > 0 THEN
mini# = 9E+14
FOR rtr = 1 TO trr
IF DistE(rtr) < mini# THEN mini# = DistE(tr): tr1 = rt(rtr)
NEXT rtr
PSET (j - n, i - m), RenkKodu(SinifNo(tr1))
PikselE(tr1) = PikselE(tr1) + 1
END IF
NEXT j, i
CLOSE
LOCATE 14, 5: PRINT " Elipsoid Metodu "
LOCATE 30, 2: PRINT SPC(75);
kol = 1: ko = 0
FOR tr = 1 TO s1
IF tr = 14 THEN ko = 1: kol = 1
LOCATE 28 + ko, kol: COLOR RenkKodu(SinifNo(tr)): PRINT PikselE(tr);
ElipsoidPikselTop = ElipsoidPikselTop + PikselE(tr)
kol = kol + 6
NEXT tr
LOCATE 30, 40: COLOR 7: PRINT "Toplam Piksel Sayısı "; ElipsoidPikselTop;
Saat2$ = TIMES
LOCATE 30, 5: COLOR 7: PRINT "Süre: "; SaatFarki(Saat1$, Saat2$); "saniye";
Yontem$ = "3ElpsMtd"

```

```

CALL DosyayaYaz(Yontem$)
a$ = INPUT$(1)
LOCATE 14, 5: PRINT SPC(20);
LOCATE 28, 1: PRINT SPC(79);
LOCATE 29, 1: PRINT SPC(79);
LOCATE 30, 2: PRINT SPC(75);
END SUB

```

```

SUB EnKucukMes

```

```

' -----
' En Küçük Mesafe yönteminin uygulanması.
' -----

```

```

Saat1$ = TIMES$
CLS
VIEW (0, 0)-(638, 420), 8, 5
DIM Dosy$(3), A1$(3), AH1$(3), Edge%(3), PikselEn(26)
DIM Dist(26) AS SINGLE, Dist1(26) AS SINGLE
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz... "; : COLOR 7
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
EnKucukPikselTop = 0
FOR i = m TO m1 + m
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn
FOR j = n TO n1 + n
FOR tr = 1 TO s1
FOR dn = 1 TO 3
Edge%(dn) = ASC(MID$(A1$(dn), j, 1))
Dist(tr) = Dist(tr) + (Edge%(dn) - Ort(dn, tr)) ^ 2
NEXT dn
Dist1(tr) = SQR(Dist(tr))
NEXT tr
mini = 9000000
FOR tr = 1 TO s1
IF Dist1(tr) < mini THEN mini = Dist1(tr): tr1 = tr
NEXT tr
PSET (j - n, i - m), RenkKodu(SinifNo(tr1))
PikselEn(tr1) = PikselEn(tr1) + 1
FOR tr = 1 TO s1
Dist(tr) = 0
NEXT tr
NEXT j, i
CLOSE

```

```
LOCATE 14, 5: PRINT " EnKüçükMesafe Metodu "  
LOCATE 30, 2: PRINT SPC(75);  
kol = 1: ko = 0  
FOR tr1 = 1 TO s1  
    IF tr1 = 14 THEN ko = 1: kol = 1  
    LOCATE 28 + ko, kol: COLOR RenkKodu(SinifNo(tr1)): PRINT PikselEn(tr1);  
    EnKucukPikselTop = EnKucukPikselTop + PikselEn(tr1)  
kol = kol + 6  
NEXT tr1  
LOCATE 30, 40: COLOR 7: PRINT "Toplam Piksel Sayısı :"; EnKucukPikselTop;  
Saat2$ = TIMES$  
LOCATE 30, 5: COLOR 7: PRINT "Süre :"; SaatFarki(Saat1$, Saat2$); "saniye";  
Yontem$ = "4EkmsMtd"  
CALL DosyayaYaz(Yontem$)  
a$ = INPUT$(1)  
LOCATE 14, 5: PRINT SPC(25);  
LOCATE 28, 1: PRINT SPC(79);  
LOCATE 29, 1: PRINT SPC(79);  
LOCATE 30, 2: PRINT SPC(75);  
END SUB
```

SUB EnYakinKomsu

' En Yakın Komşu yönteminin uygulanması.

```
Saat1$ = TIMES$
CLS
VIEW (0, 0)-(638, 420), 8, 5
DIM Dosy$(3), A1$(3), AH1$(3), Edge%(3), PikselINN(26)
DIM Dist1(26) AS SINGLE
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz... "; : COLOR 7
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
EnYakinPikselTop = 0
FOR i = m TO m1 + m
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn
FOR j = n TO n1 + n
FOR tr = 1 TO s1
FOR PikNo = 1 TO Sayi55(tr)
DistTop = 0: MiniDist = 9000000
FOR dn = 1 TO 3
Edge%(dn) = ASC(MID$(A1$(dn), j, 1))
```

```

DistTop = DistTop + ABS(Edge%(dn) - Piksel(tr, dn, PikNo))
NEXT dn
IF DistTop < MiniDist THEN MiniDist = DistTop
NEXT PikNo
Dist1(tr) = MiniDist
NEXT tr
mini = 9000000
FOR tr = 1 TO s1
  IF Dist1(tr) < mini THEN mini = Dist1(tr): tr1 = tr
NEXT tr
  PSET (j - n, i - m), RenkKodu(SinifNo(tr1))
  PikselNN(tr1) = PikselNN(tr1) + 1
FOR tr = 1 TO s1
  Dist1(tr) = 0
NEXT tr
NEXT j, i
CLOSE
LOCATE 14, 5: PRINT " EnYakinKomşu Metodu "
LOCATE 30, 2: PRINT SPC(75);
kol = 1: ko = 0
FOR tr1 = 1 TO s1
  IF tr1 = 14 THEN ko = 1: kol = 1
  LOCATE 28 + ko, kol: COLOR RenkKodu(SinifNo(tr1)): PRINT PikselNN(tr1);
  EnYakinPikselTop = EnYakinPikselTop + PikselNN(tr1)
kol = kol + 6
NEXT tr1
  LOCATE 30, 40: COLOR 7: PRINT "Toplam Piksel Sayısı :": EnYakinPikselTop;
Saat2$ = TIMES$
LOCATE 30, 5: COLOR 7: PRINT "Süre :"; SaatFarki(Saat1$, Saat2$); "saniye";
Yontem$ = "6EnykMtd"
CALL DosyayaYaz(Yontem$)
a$ = INPUT$(1)
LOCATE 14, 5: PRINT SPC(25);
LOCATE 28, 1: PRINT SPC(79);
LOCATE 29, 1: PRINT SPC(79);
LOCATE 30, 2: PRINT SPC(75);
END SUB

```

SUB gericiz

' Çerçevenin silinmesi.

```

i1 = 0
FOR i = x1 TO x2
  i1 = i1 + 1
  PSET (i, y1), a(i1)
NEXT i
j1 = 0
FOR j = y1 + 1 TO y2
  j1 = j1 + 1

```



```

PSET (x2, j), b(j1)
NEXT j
k1 = 0
FOR k = x2 - 1 TO x1 STEP -1
k1 = k1 + 1
PSET (k, y2), c(k1)
NEXT k
l1 = 0
FOR l = y2 - 1 TO y1 + 1 STEP -1
l1 = l1 + 1
PSET (x1, l), d(l1)
NEXT l
END SUB

```

SUB gezgin

'Görüntü üzerinde gezebilen çerçevenin hazırlanması ve yapılan seçim sonucunda ilgili alt prosedürün çağırılması.

```

x1 = 0: x2 = 20: y1 = 0: y2 = 20
CALL menu2          ' "menu2" prosedürünün çağırılması.
2 CALL kareciz      ' "kareciz" prosedürünün çağırılması.
DO
LOCATE 30, 55: PRINT x1 + 1; y1 + 1; x2 - x1;
DO
  bas$ = INKEY$
  LOOP WHILE bas$ = ""
  SELECT CASE bas$
  CASE CHR$(27)      'ESC - CIKIS
    CALL gericiz
  CASE CHR$(90)      'Z - ZOOM
    CALL zoom
  CASE CHR$(72)      'H - KUCUK ALAN HISTOGRAMI
    CALL histog3     ' "histog3" prosedürünün çağırılması.
  CASE CHR$(0) + CHR$(72) 'UP
    CALL gericiz
    y1 = y1 - 1: y2 = y2 - 1
    CALL kareciz
  CASE CHR$(0) + CHR$(77) 'RIGHT
    CALL gericiz
    x1 = x1 + 1: x2 = x2 + 1
    CALL kareciz
  CASE CHR$(0) + CHR$(80) 'DOWN
    CALL gericiz
    y1 = y1 + 1: y2 = y2 + 1
    CALL kareciz
  CASE CHR$(0) + CHR$(75) 'LEFT
    CALL gericiz
    x1 = x1 - 1: x2 = x2 - 1
    CALL kareciz

```



```

his(c%) = his(c%) + 1          ' Aynı renkten olan piksel sayısının hesabı.
NEXT j, i
CLOSE
LOCATE 30, 2: PRINT SPC(75);
END SUB

```

```

SUB GorZoom

```

```

' -----
' Sınıflanmış görüntünün ekrana 2 kat büyütülmüş olarak çizimi.
' -----

```

```

DIM Dosy$(3), A1$(3), AH1$(3)
DO
LOCATE 7, 40: INPUT "BANT SAYISI : ", BantSayi%
LOOP UNTIL BantSayi% = 1 OR BantSayi% = 3
IF BantSayi% = 1 THEN
LOCATE 8, 40: INPUT "DOSYA ADI : ", Dosya$
CLS : SCREEN 12
OPEN "R", #1, Dosya$, 200
FIELD #1, 200 AS AH$
FOR i = 1 TO 198
GET #1, i
a$ = AH$
FOR iei = i + 1 TO 2 + i
FOR j = 1 TO 200
IF UCASE$(RIGHT$(Dosya$, 3)) = "MTD" THEN
c% = ASC(MID$(a$, j, 1)) - 64
ELSE
c% = ASC(MID$(a$, j, 1))
c% = INT(c% / 6)
END IF
FOR jej = 1 + j TO 2 + j
PSET (j + jej, i + iei), c%
NEXT jej
NEXT j
NEXT iei
NEXT i
ELSEIF BantSayi% = 3 THEN
CLS : SCREEN 12
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
FOR i = 1 TO 198
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn

```

```

FOR iei = i + 1 TO 2 + i
FOR j = 1 TO 200
FOR dn = 1 TO 3
c1% = ASC(MID$(A1$(dn), j, 1))
ct% = ct% + c1%
NEXT dn
c% = ct% / 3
ct% = 0
c2% = INT(c% / 6 + .5)
FOR jej = 1 + j TO 2 + j
PSET (j + jej, i + iei), c2%
NEXT jej
NEXT j
NEXT iei
NEXT i
END IF
CLOSE
IF UCASE$(RIGHT$(Dosya$, 3)) = "MTD" THEN
LOCATE 26, 47: PRINT "No:"; LEFT$(Dosya$, 1)
ELSE
LOCATE 26, 45: PRINT "Band-"; MID$(Dosya$, 5, 1)
END IF
LOCATE 26, 1: PRINT "0"
LOCATE 26, 9: PRINT "1 km"
LINE (3, 416)-(4, 426), 15, BF
LINE (3, 420)-(68, 421), 15, BF
LINE (67, 416)-(68, 426), 15, BF
LINE (410, 10)-(405, 15), 15
LINE (411, 10)-(416, 15), 15
LINE (410, 10)-(411, 51), 15, BF
LOCATE 3, 53: PRINT "N";
DO
da$ = INPUT$(1)
LOOP UNTIL ASC(da$) = 27
SCREEN 0
END SUB

```

SUB histog1

' Tm grntnn histogram deęerlerinin grntlenmesi.

HisTop = 0: Sayi = 0: j = 0

FOR i = 0 TO 255

j = j + 1

LOCATE j, 1: PRINT i, his(i)

IF his(i) > 0 THEN

Sayi = Sayi + 1

HisTop = HisTop + his(i) * i

END IF

IF j > 20 THEN

j = 0

'Mevcut parlaklik sayisi.

'Parlaklik deęerlerinin toplamı.

```

DO
  LOOP UNTIL INKEY$ <> ""
  CLS
END IF
NEXT i
i = 0
PikSay = (m1 - m + 1) * (n1 - n + 1)
OrtHis = HisTop / PikSay
Ts = 0: Tss = 0
FOR i = 0 TO 255
  IF his(i) > 0 THEN
    Sapma = his(i) - OrtHis           'Sapmaların hesabı
    Ts = Ts + ABS(Sapma)              'Sapmaların toplanması
    Tss = Tss + Sapma ^ 2             'Sapmaların karelerinin toplamı
  END IF
NEXT i
Os = Ts / Sayi                       'Ortalama sapmanın hesabı
Ss = SQR(Tss / (Sayi - 1))           'Standart sapmanın hesabı
DO
  LOOP UNTIL INKEY$ <> ""
CLS
END SUB

```

```

SUB histog2

```

```

'-----
' Tüm görüntünün histogramının grafiksel yorumu.
'-----

```

```

DIM sakla%(32765)
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz..."; : COLOR 7
nn = 0
FOR y = 294 TO 411
  FOR x = 2 TO 274
    nn = nn + 1
    sakla%(nn) = POINT(x, y)
  NEXT x, y
  VIEW (3, 295)-(273, 410), 8, 5
  LINE (5, 110)-(265, 110), 15
  LOCATE 20, 2: PRINT "0": LOCATE 20, 32: PRINT "255"
  FOR i = 0 TO 255
    IF his(i) <> 0 THEN
      LINE (5 + i, 110)-(5 + i, 110 - his(i) / 100), 2
    END IF
  NEXT i
  LOCATE 30, 1: PRINT SPC(75);
  DO
  LOOP UNTIL INKEY$ = CHR$(27)
  VIEW
  nn = 0
  FOR y = 294 TO 411
    FOR x = 2 TO 274

```

```

nn = nn + 1
PSET (x, y), sakla%(nn)
NEXT x, y
ERASE sakla%
END SUB

```

```

SUB histog3

```

```

'-----
' 3.bantta çerçeve aracılığıyla seçilen bir alana ait piksellerin histogramının ve diğer istatistiksel
' değerlerinin hesabı.
'-----

```

```

Tc3 = 0: Sayi3 = 0: Ts3 = 0: Tss3 = 0: l = 0

```

```

DIM sakla%(32765)

```

```

Dosya$ = "Ada73.y1"

```

```

LOCATE 30, 1: PRINT SPC(75);

```

```

LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz..."; : COLOR 7

```

```

OPEN "R", #1, Dosya$, 200

```

```

FIELD #1, 200 AS AHS

```

```

FOR i = m + y1 TO m + y2 - 2

```

```

GET #1, i

```

```

a$ = AHS

```

```

FOR j = n + x1 TO n + x2 - 2

```

```

c% = ASC(MID$(a$, j, 1))

```

```

Sayi3 = Sayi3 + 1

```

```

Tc3 = Tc3 + c%

```

```

his3(c%) = his3(c%) + 1

```

```

NEXT j, i

```

```

Orthis3 = Tc3 / Sayi3

```

```

FOR i = m + y1 TO m + y2 - 2

```

```

GET #1, i

```

```

a$ = AHS

```

```

FOR j = n + x1 TO n + x2 - 2

```

```

c% = ASC(MID$(a$, j, 1))

```

```

    Sapma = c% - Orthis3

```

```

    Ts3 = Ts3 + ABS(Sapma)

```

```

    Tss3 = Tss3 + Sapma ^ 2

```

```

NEXT j, i

```

```

CLOSE

```

```

Os3 = Ts3 / Sayi3

```

```

Ss3 = SQR(Tss3 / (Sayi3 - 1))

```

```

IF x2 < 320 THEN

```

```

    kayX = 330: kayL = 41

```

```

ELSE

```

```

    kayX = 0: kayL = 0

```

```

END IF

```

```

nn = 0

```

```

FOR y = 5 TO 122

```

```

FOR x = 2 + kayX TO 274 + kayX

```

```

nn = nn + 1

```

```

sakla%(nn) = POINT(x, y)

```

'Piksel sayısının hesabı

'Parlaklık değerlerinin toplamı

'Sapmaların hesabı.

'Sapmaların toplanması.

'Sapmaların karelerinin toplamı.

'Ortalama sapmanın hesabı.

'Standart sapmanın hesabı.

```

NEXT x, y
VIEW (3 + kayX, 6)-(273 + kayX, 121), 8, 5
LINE (5, 110)-(265, 110), 15
LOCATE 2, 2 + kayL: PRINT "0": LOCATE 2, 32 + kayL: PRINT "255"
LOCATE 3, 2 + kayL: PRINT Orthis3; Os3; Ss3
FOR i = 0 TO 255
IF his3(i) <> 0 THEN
LINE (5 + i, 110)-(5 + i, 110 - his3(i)), 2
END IF
NEXT i
CALL histog4                                ' "histog4" prosedürünün çağırılması.
VIEW
nn = 0
FOR y = 5 TO 122
FOR x = 2 + kayX TO 274 + kayX
nn = nn + 1
PSET (x, y), sakla%(nn)
NEXT x, y
ERASE his3, sakla%
CALL menu2                                ' "menu2" prosedürünün çağırılması.
END SUB

```

```

SUB histog4

```

```

' -----
' 4.bantta çerçeve aracılığıyla seçilen bir alana ait piksellerin histogramının ve diğer istatistiksel
' değerlerinin hesabı.
' -----

```

```

Tc4 = 0: Sayi4 = 0: Ts4 = 0: Tss4 = 0: l = 0
DIM sakla1%(32765)
VIEW
ERASE his3
Dosya$ = "Ada74.y1"
OPEN "R", #1, Dosya$, 200
FIELD #1, 200 AS AH$
FOR i = m + y1 TO m + y2 - 2
GET #1, i
a$ = AH$
FOR j = n + x1 TO n + x2 - 2
c% = ASC(MID$(a$, j, 1))
Sayi4 = Sayi4 + 1                                'Piksel sayisinin hesabi
Tc4 = Tc4 + c%                                'Parlaklik degerlerinin toplami
his3(c%) = his3(c%) + 1
NEXT j, i
Orthis4 = Tc4 / Sayi4
FOR i = m + y1 TO m + y2 - 2
GET #1, i
a$ = AH$
FOR j = n + x1 TO n + x2 - 2
c% = ASC(MID$(a$, j, 1))
Sapma = c% - Orthis4                                'Sapmaların hesabı

```



```

        Ts4 = Ts4 + ABS(Sapma)           'Sapmaların toplanması
        Tss4 = Tss4 + Sapma ^ 2         'Sapmaların karelerinin toplamı
    NEXT j, i
    CLOSE
    Os4 = Ts4 / Sayi4                   'Ortalama sapmanın hesabı
    Ss4 = SQR(Tss4 / (Sayi4 - 1))       'Standart sapmanın hesabı
    IF x2 < 320 THEN
        kayX = 330: kayL = 41
    ELSE
        kayX = 0: kayL = 0
    END IF
    nn = 0
    FOR y = 135 TO 252
        FOR x = 2 + kayX TO 274 + kayX
            nn = nn + 1
            sakla1%(nn) = POINT(x, y)
        NEXT x, y
        VIEW (3 + kayX, 136)-(273 + kayX, 251), 8, 5
        LINE (5, 110)-(265, 110), 15
        LOCATE 10, 2 + kayL: PRINT "0": LOCATE 10, 32 + kayL: PRINT "255"
        LOCATE 11, 2 + kayL: PRINT Orthi4; Os4; Ss4
        FOR i = 0 TO 255
            IF his3(i) <> 0 THEN
                LINE (5 + i, 110)-(5 + i, 110 - his3(i)), 2
            END IF
        NEXT i
        CALL histog5                     ' "histog5" prosedürünün çağırılması.
        VIEW
        nn = 0
        FOR y = 135 TO 252
            FOR x = 2 + kayX TO 274 + kayX
                nn = nn + 1
                PSET (x, y), sakla1%(nn)
            NEXT x, y
            ERASE his3, sakla1%
        END SUB

```

SUB histog5

' 5.bantta çerçeve aracılığıyla seçilen bir alana ait piksellerin histogramının ve diğer istatistiksel
' değerlerinin hesabı.

```

Tc5 = 0: Sayi5 = 0: Ts5 = 0: Tss5 = 0: l = 0
DIM sakla2%(32765)
VIEW
ERASE his3
Dosya$ = "Ada75.y1"
OPEN "R", #1, Dosya$, 200
FIELD #1, 200 AS AHS

```

```

FOR i = m + y1 TO m + y2 - 2
  GET #1, i
  a$ = AHS
  FOR j = n + x1 TO n + x2 - 2
    c% = ASC(MID$(a$, j, 1))
    Sayi5 = Sayi5 + 1
    Tc5 = Tc5 + c%
    his3(c%) = his3(c%) + 1
  NEXT j, i
  Orthi5 = Tc5 / Sayi5
  FOR i = m + y1 TO m + y2 - 2
    GET #1, i
    a$ = AHS
    FOR j = n + x1 TO n + x2 - 2
      c% = ASC(MID$(a$, j, 1))
      Sapma = c% - Orthi5
      Ts5 = Ts5 + ABS(Sapma)
      Tss5 = Tss5 + Sapma ^ 2
    NEXT j, i
  CLOSE
  Os5 = Ts5 / Sayi5
  Ss5 = SQR(Tss5 / (Sayi5 - 1))
  IF x2 < 320 THEN
    kayX = 330: kayL = 41
  ELSE
    kayX = 0: kayL = 0
  END IF
  nn = 0
  FOR y = 265 TO 382
    FOR x = 2 + kayX TO 274 + kayX
      nn = nn + 1
      sakla2%(nn) = POINT(x, y)
    NEXT x, y
    VIEW (3 + kayX, 266)-(273 + kayX, 381), 8, 5
    LINE (5, 110)-(265, 110), 15
    LOCATE 18, 2 + kayL: PRINT "0": LOCATE 18, 32 + kayL: PRINT "255"
    LOCATE 19, 2 + kayL: PRINT Orthi5; Os5; Ss5
    FOR i = 0 TO 255
      IF his3(i) <> 0 THEN
        LINE (5 + i, 110)-(5 + i, 110 - his3(i)), 2
      END IF
    NEXT i
    LOCATE 30, 1: PRINT SPC(75);
    CALL OrnekAlan
    DO
    LOOP UNTIL INKEY$ = CHR$(27)
    VIEW
    nn = 0
    FOR y = 265 TO 382
    FOR x = 2 + kayX TO 274 + kayX
    nn = nn + 1

```

'Piksel sayisinin hesabi
'Parlaklik degerlerinin toplami

'Sapmalarin hesabi
'Sapmalarin toplanmasi
'Sapmalarin karelerinin toplami

'Ortalama sapmanin hesabi
'Standart sapmanin hesabi

' "OrnekAlan" prosedürünün çağırılması.

```
PSET (x, y), 8 'sakla2%(nn)
NEXT x, y
ERASE his3, sakla2%
END SUB
```

```
SUB kareviz
```

```
' -----
' Çerçevenin çizilmesi.
' -----
```

```
il = 0
FOR i = x1 TO x2
il = il + 1
a(il) = POINT(i, y1)
PSET (i, y1), 15
NEXT i
jl = 0
FOR j = y1 + 1 TO y2
jl = jl + 1
b(jl) = POINT(x2, j)
PSET (x2, j), 15
NEXT j
kl = 0
FOR k = x2 - 1 TO x1 STEP -1
kl = kl + 1
c(kl) = POINT(k, y2)
PSET (k, y2), 15
NEXT k
ll = 0
FOR l = y2 - 1 TO y1 + 1 STEP -1
ll = ll + 1
d(ll) = POINT(x1, l)
PSET (x1, l), 15
NEXT l
END SUB
```

```
SUB Kovaryans
```

```
' -----
' Max. Benzerlik yöntemi için kovaryans matris ,determinantı ve tersinin hesaplanması.
' -----
```

```
DIM Si(3, 3), KofSi(3, 3), AdjSi(3, 3), Dosy$(3), A1$(3), AH1$(3)
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
Sayi55(s1) = 0
FOR i = m + y1 TO m + y2 - 2
GET #dn, i
```

```

A1$(dn) = AH1$(dn)
FOR j = n + x1 TO n + x2 - 2
c% = ASC(MID$(A1$(dn), j, 1))
Sayi55(s1) = Sayi55(s1) + 1
Piksel(s1, dn, Sayi55(s1)) = c%
NEXT j, i
NEXT dn
CLOSE
FOR j = 1 TO 3
FOR k = 1 TO 3
Top = 0
FOR l = 1 TO Sayi55(s1)
Top = Top + (Piksel(s1, j, l) - Ort(j, s1)) * (Piksel(s1, k, l) - Ort(k, s1))
NEXT l
Si(j, k) = Top / (Sayi55(s1) - 1)
NEXT k, j
' Sigmanın determinanti
detSigma(s1) = Si(1, 1) * (Si(2, 2) * Si(3, 3) - Si(3, 2) * Si(2, 3)) - Si(1, 2) * (Si(2, 1) * Si(3, 3) - Si(3, 1) * Si(2, 3)) + Si(1, 3) * (Si(2, 1) * Si(3, 2) - Si(3, 1) * Si(2, 2))
' Matrisin tersi
IF detSigma(s1) <= 0 THEN BEEP: BEEP: s1 = s1 - 1: EXIT SUB
' Kofaktor ve Adjoint hesabi
m9 = 2: n9 = 3
FOR i = 1 TO 3
m9 = m9 - 1: n9 = n9 - 1: k9 = 2: l9 = 3
FOR j = 1 TO 3
k9 = k9 - 1: l9 = l9 - 1
AdjSi(i, j) = (Si(i + m9, j + k9) * Si(i + n9, j + l9) - Si(i + n9, j + k9) * Si(i + m9, j + l9)) * (-1) ^ (i + j)
IF j = 1 THEN k9 = k9 - 1
IF j = 2 THEN l9 = l9 - 1
NEXT j
IF i = 1 THEN m9 = m9 - 1
IF i = 2 THEN n9 = n9 - 1
NEXT i
' Matrisin tersi
FOR i = 1 TO 3
FOR j = 1 TO 3
IF detSigma(s1) > 0 THEN
TersSi(s1, i, j) = AdjSi(j, i) / detSigma(s1)
END IF
NEXT j
NEXT i
END SUB

SUB KutuMetodu
' -----
' Kutu (paralel piped) yönteminin uygulanması.
' -----

Saat1$ = TIMES
CLS

```

```

VIEW (0, 0)-(638, 420), 8, 5
DIM Dosy$(3), A1$(3), AH1$(3), Edge%(3), PikselK(26)
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz... "; : COLOR 7
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
KutuPikselTop = 0
FOR i = m TO m1 + m
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn
FOR j = n TO n1 + n
FOR tr = 1 TO s1
zz% = 0
FOR dn = 1 TO 3
Edge%(dn) = ASC(MID$(A1$(dn), j, 1))
IF Edge%(dn) >= kenar(dn, tr) AND Edge%(dn) <= kenar(dn + 3, tr) THEN
zz% = zz% + 1
END IF
NEXT dn
IF zz% = 3 THEN
PSET (j - n, i - m), RenkKodu(SinifNo(tr))
PikselK(tr) = PikselK(tr) + 1
zz% = 0
EXIT FOR
END IF
NEXT tr
NEXT j, i
CLOSE
LOCATE 14, 5: PRINT " Kutu Metodu "
LOCATE 30, 2: PRINT SPC(75);
kol = 1: ko = 0
FOR tr = 1 TO s1
IF tr = 14 THEN ko = 1: kol = 1
LOCATE 28 + ko, kol: COLOR RenkKodu(SinifNo(tr)): PRINT PikselK(tr);
KutuPikselTop = KutuPikselTop + PikselK(tr)
kol = kol + 6
NEXT tr
LOCATE 30, 40: COLOR 7: PRINT "Toplam Piksel Sayısı :"; KutuPikselTop;
Saat2$ = TIMES
LOCATE 30, 5: COLOR 7: PRINT "Süre :"; SaatFarki(Saat1$, Saat2$); "saniye";
Yontem$ = "2KutuMtd"
CALL DosyayaYaz(Yontem$)
a$ = INPUT$(1)
LOCATE 14, 5: PRINT SPC(20);
LOCATE 28, 1: PRINT SPC(79);

```

```
LOCATE 29, 1: PRINT SPC(79);
LOCATE 30, 2: PRINT SPC(75);
END SUB
```

```
SUB MaxBenzerlik
```

```
'-----
' Maksimum Benzerlik yönteminin uygulanması.
'-----
```

```
Saat1$ = TIMES
CLS
VIEW (0, 0)-(638, 420), 8, 5
DIM Dosy$(3), A1$(3), AH1$(3), Edge%(3), PikselMax(26)
DIM Dist2(26) AS DOUBLE, t5(3), Pik(3)
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz... "; : COLOR 7
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
MaxBenPikselTop = 0
FOR i = m TO m1 + m
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn
FOR j = n TO n1 + n
FOR tr = 1 TO s1
FOR dn = 1 TO 3
Edge%(dn) = ASC(MID$(A1$(dn), j, 1))
Pik(dn) = Edge%(dn) - Ort(dn, tr)
NEXT dn
t5(1) = 0: t5(2) = 0: t5(3) = 0
FOR j5 = 1 TO 3
FOR k5 = 1 TO 3
t5(j5) = t5(j5) + Pik(k5) * TersSi(tr, k5, j5)
NEXT k5, j5
t6 = 0
FOR j5 = 1 TO 3
t6 = t6 + Pik(j5) * t5(j5)
NEXT j5
IF detSigma(tr) > 0 THEN
'Dist2(tr) = EXP(-(1 / 2) * t6) / ((2 * Pi) ^ (1.5) * SQR(detSigma(tr)))
'Dist2(tr) = -1.5 * LOG(2 * Pi) - (1 / 2) * LOG(detSigma(tr)) - (1 / 2) * t6
Dist2(tr) = LOG(1 / s1) - (1 / 2) * LOG(detSigma(tr)) - (1 / 2) * t6
END IF
NEXT tr
max# = -9E+14
FOR tr = 1 TO s1
```

```

    IF Dist2(tr) > max# THEN max# = Dist2(tr): tr1 = tr
NEXT tr
    PSET (j - n, i - m), RenkKodu(SinifNo(tr1))
    PikselMax(tr1) = PikselMax(tr1) + 1
NEXT j, i
CLOSE
LOCATE 14, 5: PRINT " Max.Benzerlik Metodu "
LOCATE 30, 2: PRINT SPC(75);
kol = 1: ko = 0
FOR tr1 = 1 TO s1
    IF tr1 = 14 THEN ko = 1: kol = 1
    LOCATE 28 + ko, kol: COLOR RenkKodu(SinifNo(tr1)): PRINT PikselMax(tr1);
    MaxBenPikselTop = MaxBenPikselTop + PikselMax(tr1)
    kol = kol + 6
NEXT tr1
    LOCATE 30, 40: COLOR 7: PRINT "Toplam Piksel Sayısı :"; MaxBenPikselTop;
Saat2$ = TIMES$
LOCATE 30, 5: COLOR 7: PRINT "Süre :"; SaatFarki(Saat1$, Saat2$); "saniye";
Yontem$ = "5MxBeMtd"
CALL DosyayaYaz(Yontem$)
a$ = INPUT$(1)
LOCATE 14, 5: PRINT SPC(25);
LOCATE 28, 1: PRINT SPC(79);
LOCATE 29, 1: PRINT SPC(79);
LOCATE 30, 2: PRINT SPC(75);
END SUB

```

SUB menu

' Ana menünün hazırlanması ve ana menüden yapılan seçim sonucunda ilgili alt prosedürlerin çağırılması.

```

DIM m$(6)
DO
CLS
m$(1) = " Görüntü  "
m$(2) = " Histogram "
m$(3) = " 255 Renk  "
m$(4) = " Sınıf Hist "
m$(5) = " Gör Zoom  "
m$(6) = " Çıkış    "
FOR i = 1 TO 6
LOCATE 2 * i + 3, 10: COLOR 5, 2: PRINT m$(i): COLOR 7, 0
NEXT i
FOR i = 1 TO 6
LOCATE 2 * i + 3, 10: COLOR 7, 5: PRINT m$(i): COLOR 7, 0
DO
10 a$ = INKEY$
LOOP UNTIL a$ <> ""
IF ASC(a$) = 32 THEN
    LOCATE 2 * i + 3, 10: COLOR 5, 2: PRINT m$(i): COLOR 7, 0

```



```

ELSEIF ASC(a$) = 13 THEN
  EXIT FOR
ELSE GOTO 10
END IF
IF i = 6 THEN i = 0
NEXT i
SELECT CASE i
CASE 1
  CALL Boyutlar          ' "Boyutlar" prosedürünün çağırılması.
  CALL Menu1             ' "menu1" prosedürünün çağırılması.
CASE 2
  CALL histog1           ' "histog1" prosedürünün çağırılması.
CASE 3
  CALL Renk255           ' "histog1" prosedürünün çağırılması.
CASE 4
  CALL SinifHistog       ' "Sinifhistog" prosedürünün çağırılması.
CASE 5
  CALL GorZoom           ' "GorZoom" prosedürünün çağırılması.
CASE ELSE
  SCREEN 0: CLS
  END                    ' Program sonu.
END SELECT
LOOP
END SUB

```

SUB Menu1

' Alt menünün hazırlanması ve bu menüden yapılan seçim sonucunda ilgili alt prosedürlerin çağırılması.

```

DO
LOCATE 30, 1: COLOR 14: PRINT "O"; : COLOR 3: PRINT "rneklemesi ";
LOCATE 30, 15: COLOR 14: PRINT "H"; : COLOR 3: PRINT "istogram";
LOCATE 30, 30: COLOR 14: PRINT "Y"; : COLOR 3: PRINT "oğ.Kesimi";
LOCATE 30, 45: COLOR 14: PRINT "S"; : COLOR 3: PRINT "inifla";
LOCATE 30, 77: COLOR 14: PRINT "E"; : COLOR 3: PRINT "sc"; : COLOR 7
sec$ = INPUT$(1)
SELECT CASE ASC(sec$)
CASE 79
  CALL gezgin           ' "gezgin" prosedürünün çağırılması.
CASE 72
  LOCATE 30, 1: PRINT SPC(75);
  CALL histog2          ' "histog2" prosedürünün çağırılması.
CASE 89
  LOCATE 30, 1: PRINT SPC(75);
  CALL YogKesimi        ' "YogKesimi" prosedürünün çağırılması.
CASE 83
  CALL Sinifla          ' "Sinifla" prosedürünün çağırılması.
CASE 27

```

```

SCREEN 0
END SELECT
LOOP UNTIL ASC(sec$) = 27
END SUB

```

```

SUB menu2

```

```

' -----
' Alt menünün hazırlanması.
' -----

```

```

LOCATE 30, 1: PRINT SPC(75);
LOCATE 30, 10: COLOR 14: PRINT "Z"; : COLOR 3: PRINT "oom";
LOCATE 30, 28: COLOR 14: PRINT "H"; : COLOR 3: PRINT "istogram"; : COLOR 7
END SUB

```

```

SUB OrnekAlan

```

```

' -----
' Üçgen yöntemi için üçgen alanlarının hesabı ve diğer yöntemler için kullanılacak değerlerin
' hesaplanacağı ilgili alt prosedürlerin çağırılması.
' -----

```

```

s1 = s1 + 1
Ort(1, s1) = Orthi3: Ss(1, s1) = Ss3
Ort(2, s1) = Orthi4: Ss(2, s1) = Ss4
Ort(3, s1) = Orthi5: Ss(3, s1) = Ss5
kenar(1, s1) = Orthi3 - RR * Ss3
kenar(2, s1) = Orthi4 - RR * Ss4
kenar(3, s1) = Orthi5 - RR * Ss5
Cevre = (kenar(1, s1) + kenar(2, s1) + kenar(3, s1)) / 2
AlanK1 = Cevre * (Cevre - kenar(1, s1)) * (Cevre - kenar(2, s1)) * (Cevre - kenar(3, s1))
Alan1(s1) = SQR(ABS(AlanK1))
kenar(4, s1) = Orthi3 + RR * Ss3
kenar(5, s1) = Orthi4 + RR * Ss4
kenar(6, s1) = Orthi5 + RR * Ss5
Cevre = (kenar(4, s1) + kenar(5, s1) + kenar(6, s1)) / 2
AlanK2 = Cevre * (Cevre - kenar(4, s1)) * (Cevre - kenar(5, s1)) * (Cevre - kenar(6, s1))
Alan2(s1) = SQR(ABS(AlanK2))
LOCATE 30, 60: PRINT s1;
CALL Kovaryans ' "Kovaryans" prosedürünün çağırılması.
CALL SinifNoHes ' "SinifNoHes" prosedürünün çağırılması.
LOCATE 30, 40: PRINT "memory="; FRE(-1);
END SUB

```

```

SUB Renk255

```

```

' -----
' Üç bantta ham görüntünün 320*200 çözünürlüklü ekranda, 255 tane renk sayısı kullanılarak çizilmesi.
' -----

```

```

DIM Dosy$(3), A1$(3), AH1$(3)
LOCATE 9, 40: INPUT "İLK SATIR NO : ", m
LOCATE 10, 40: INPUT "SATIR SAYISI : ", m1

```

```

LOCATE 11, 40: INPUT "İLK SÜTUN NO : ", n
LOCATE 12, 40: INPUT "SÜTUN SAYISI : ", n1
IF m1 > 200 THEN m1 = 200
IF n1 > 320 THEN n1 = 320
CLS : SCREEN 13
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200      ' Dosyaların açılması.
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
FOR i = m TO m1 + m                ' Belirtilen satır sayısı kez
                                  ' her dosyanın sırası gelen
                                  ' kaydının (satır) okunması.
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn
FOR j = n TO n1 + n                ' Her satırın belirtilen
                                  ' sütunundan itibaren karakterler
FOR dn = 1 TO 3                    ' birer birer okunarak ASCII
c1% = ASC(MID$(A1$(dn), j, 1))     ' değerlerinin bulunup ortalamasının
ct% = ct% + c1%                    ' hesaplanması ve bu değere
NEXT dn                            ' karşılık gelen renk ile
c% = ct% / 3                       ' ekranın aynı adresindeki
ct% = 0                            ' pikselin boyanması.
PSET (j - n, i - m), c%
NEXT j, i
CLOSE
DO
sec$ = INPUT$(1)
LOOP UNTIL ASC(sec$) = 27
SCREEN 12
SCREEN 0
END SUB

```

SUB RenkBilesimi

' Üç banttaki görüntünün ekranda çizilmesi.

```

DIM Dosy$(3), A1$(3), AH1$(3)
ERASE his
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz... "; : COLOR 7
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200      ' Dosyaların açılması.
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
FOR i = m TO m1 + m                ' Belirtilen satır sayısı kez

```

```

FOR dn = 1 TO 3
    GET #dn, i
    A1$(dn) = AH1$(dn)
NEXT dn
FOR j = n TO n1 + n
    FOR dn = 1 TO 3
        c1% = ASC(MID$(A1$(dn), j, 1))
        ct% = ct% + c1%
    NEXT dn
    c% = ct% / 3
    ct% = 0
    c2% = INT(c% / 6 + .5)
    PSET (j - n, i - m), c2%
    his(c%) = his(c%) + 1
NEXT j, i
CLOSE
LOCATE 30, 2: PRINT SPC(75);
END SUB

```

' her dosyanın sırası gelen
' kaydının (satır) okunması.

' Her satırın belirtilen
' sütunundan itibaren karakterler
' birer birer okunarak ASCII
' değerlerinin bulunup ortalamasının
' hesaplanması ve bu değerin 6'ya
' bölünerek karşılık gelen renk ile
' ekranın aynı adresindeki
' pikselin boyanması.

' Aynı renkten olan piksel sayısının hesabı.

```

FUNCTION SaatFarki (Saat1$, Saat2$)

```

' Sınıflandırma yönteminin uygulanması esnasında geçen sürenin hesaplanması.

```

St1% = VAL(LEFT$(Saat1$, 2)): Dk1% = VAL(MID$(Saat1$, 4, 2))
Sn1% = VAL(RIGHT$(Saat1$, 2))
St2% = VAL(LEFT$(Saat2$, 2)): Dk2% = VAL(MID$(Saat2$, 4, 2))
Sn2% = VAL(RIGHT$(Saat2$, 2))
IF (St2% - St1%) < 0 THEN St2% = St2% + 24
IF Sn2% < Sn1% THEN
    Dk2% = Dk2% - 1: Sn2% = Sn2% + 60
END IF
tt1% = Sn2% - Sn1%
IF Dk2% < Dk1% THEN
    St2% = St2% - 1: Dk2% = Dk2% + 60
END IF
tt2% = (Dk2% - Dk1%) * 60
tt3 = (St2% - St1%) * 3600
SaatFarki = tt1% + tt2% + tt3
END FUNCTION

```

```

SUB SinifHistog

```

' Sınıflanmış görüntüdeki piksellerin histogram değerlerinin bir dosyaya yazılması.

```

DIM SHist(256)
FOR tr = 1 TO s1
    FOR dn = 1 TO 3
        tr$ = LTRIM$(RTRIM$(STR$(tr)))

```

```

dn$ = LTRIM$(RTRIM$(STR$(dn)))
File$ = "Hist" + tr$ + "-" + dn$
OPEN "o", #7, File$
FOR P = 0 TO 255
  SHist(P) = 0
NEXT P
FOR Psayi = 1 TO Sayi55(tr)
  c% = Piksel(tr, dn, Psayi)
  SHist(c%) = SHist(c%) + 1
NEXT Psayi
FOR P = 0 TO 255
  WRITE #7, P, SHist(P)
NEXT P
CLOSE #7
NEXT dn
NEXT tr
END SUB

```

SUB Sinifla

' Sınıflandırma yöntemlerini içeren bir alt menü ve bu menüden seçilecek yöntemle ilgili alt prosedürün çağırılması.

```

LOCATE 30, 1: PRINT SPC(75);
DO
LOCATE 30, 1: COLOR 14: PRINT "U"; : COLOR 3: PRINT "cgenMetodu";
LOCATE 30, 15: COLOR 14: PRINT "K"; : COLOR 3: PRINT "utuMetodu";
LOCATE 30, 27: COLOR 14: PRINT "E"; : COLOR 3: PRINT "lipsoid";
LOCATE 30, 40: COLOR 14: PRINT "M"; : COLOR 3: PRINT "inMesafe";
LOCATE 30, 52: COLOR 3: PRINT "Max"; : COLOR 14: PRINT "B"; : COLOR 3: PRINT "enzerlik";
LOCATE 30, 65: COLOR 14: PRINT "N"; : COLOR 3: PRINT "Komsu";
sep$ = INPUT$(1)
LOCATE 30, 1: PRINT SPC(75);
SELECT CASE ASC(sep$)
CASE 85
  CALL UcgenMetodu
CASE 75
  CALL KutuMetodu
CASE 69
  CALL Elipsoid
CASE 77
  CALL EnKucukMes
CASE 66
  CALL MaxBenzerlik
CASE 78
  CALL EnYakinKomsu
END SELECT
LOOP UNTIL ASC(sep$) = 27
LOCATE 30, 1: PRINT SPC(75);
END SUB

```

' Menüden yapılacak seçime göre

' "UcgenMetodu" prosedürünün çağırılması.

' "KutuMetodu" prosedürünün çağırılması.

' "Elipsoid" prosedürünün çağırılması.

' "EnKucukMes" prosedürünün çağırılması.

' "MaxBenzerlik" prosedürünün çağırılması.

' "EnYakinKomsu" prosedürünün çağırılması.

SUB SinifNoHes

' Seçilen sınıflara verilecek numara ve renk seçimi.

```
LOCATE 30, 2, 1: PRINT "SINIF NO :";
a$ = INPUT$(1)
SinifNo(s1) = VAL(a$)
LOCATE 30, 12: PRINT SinifNo(s1);
DO
FOR renk = 1 TO 15
LOCATE 30, 16: COLOR renk: PRINT " "; renk; : COLOR 7
22 a$ = INKEY$: IF a$ = "" THEN 22
IF ASC(a$) = 13 OR ASC(a$) = 32 THEN 23 ELSE 22
23 IF ASC(a$) = 13 THEN RenkKodu(SinifNo(s1)) = renk: EXIT DO
NEXT renk
LOOP
END SUB
```

SUB UcgenMetodu

' Üçgen yönteminin uygulanması.

```
Saat1$ = TIME$
CLS
VIEW (0, 0)-(638, 420), 8, 5
DIM Dosy$(3), A1$(3), AH1$(3), Edge%(3), PikselU(26), DistU(26), rt(26)
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz... "; : COLOR 7
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
UcgenPikselTop = 0
FOR i = m TO m1 + m
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn
UcgenPikselTop = 0
FOR j = n TO n1 + n
Cev = 0
FOR dn = 1 TO 3
Edge%(dn) = ASC(MID$(A1$(dn), j, 1))
Cev = Cev + Edge%(dn)
NEXT dn
Cev = Cev / 2
Ala = Cev * (Cev - Edge%(1)) * (Cev - Edge%(2)) * (Cev - Edge%(3))
```

```

Alan = SQR(ABS(Ala))
trr = 0
FOR tr = 1 TO s1
IF Alan >= Alan1(tr) AND Alan <= Alan2(tr) THEN
    trr = trr + 1: rt(trr) = tr
    DistU(trr) = (Alan1(tr) + Alan2(tr)) / 2 - Alan
END IF
NEXT tr
IF trr > 0 THEN
mini# = 9E+14
FOR rtr = 1 TO trr
    IF DistU(rtr) < mini# THEN mini# = DistU(rtr): tr1 = rt(rtr)
NEXT rtr
    PSET (j - n, i - m), RenkKodu(SinifNo(tr1))
    PikselU(tr1) = PikselU(tr1) + 1
END IF
NEXT j, i
CLOSE
LOCATE 14, 5: PRINT " Üçgen Metodu "
LOCATE 30, 2: PRINT SPC(75);
kol = 1: ko = 0
FOR tr = 1 TO s1
    IF tr = 14 THEN ko = 1: kol = 1
    LOCATE 28 + ko, kol: COLOR RenkKodu(SinifNo(tr)): PRINT PikselU(tr);
    UcgenPikselTop = UcgenPikselTop + PikselU(tr)
    kol = kol + 6
NEXT tr
    LOCATE 30, 40: COLOR 7: PRINT "Toplam Piksel Sayısı :"; UcgenPikselTop;
    Saat2$ = TIMES$
    LOCATE 30, 5: COLOR 7: PRINT "Süre :"; SaatFarki(Saat1$, Saat2$); "saniye";
    Yontem$ = "1UcgMtd"
    CALL DosyayaYaz(Yontem$)
    a$ = INPUT$(1)
    LOCATE 14, 5: PRINT SPC(20);
    LOCATE 28, 1: PRINT SPC(79);
    LOCATE 29, 1: PRINT SPC(79);
    LOCATE 30, 2: PRINT SPC(75);
END SUB

```

SUB YogKesimi

' Yoğunluk Kesimi işleminin yapılması.

```

FOR i = 0 TO 255
    IF his(i) > 0 AND his(i + 1) > 0 THEN
        min = i
        EXIT FOR
    END IF
NEXT i
FOR j = min TO 255

```

' 0 ile 255 aralığında
' minimum parlaklık
' değerinin bulunması.


```

IF his(j) = 0 AND his(j + 1) = 0 THEN      ' maksimum parlaklık
    max = j - 1                            ' değerinin bulunması.
EXIT FOR
END IF
NEXT j
bol = INT((max - min + 1) / 16)
LOCATE 30, 40: PRINT min; max; bol;
IF Bs% = 1 THEN
LOCATE 30, 2: INPUT ; "DOSYA ADI  : ", Dosya$
LOCATE 30, 2: PRINT SPC(75);
LOCATE 30, 2: COLOR 3: PRINT Dosya$; " yaziliyor..."; : COLOR 7
OPEN "R", #1, Dosya$, 200
FIELD #1, 200 AS AH$
FOR i = m TO m1 + m
GET #1, i
a$ = AH$
FOR j = n TO n1 + n
c% = ASC(MID$(a$, j, 1))
c2% = CINT(c% / bol) - 5
PSET (j - n, i - m), c2%
NEXT j, i
ELSEIF Bs% = 3 THEN
DIM Dosy$(3), A1$(3), AH1$(3)
LOCATE 30, 2: COLOR 3: PRINT " Bekleyiniz... "; : COLOR 7
Dosy$(1) = "Ada73.y1"
Dosy$(2) = "Ada74.y1"
Dosy$(3) = "Ada75.y1"
FOR dn = 1 TO 3
OPEN "R", #dn, Dosy$(dn), 200
FIELD #dn, 200 AS AH1$(dn)
NEXT dn
FOR i = m TO m1 + m
FOR dn = 1 TO 3
GET #dn, i
A1$(dn) = AH1$(dn)
NEXT dn
FOR j = n TO n1 + n
FOR dn = 1 TO 3
c1% = ASC(MID$(A1$(dn), j, 1))
ct% = ct% + c1%
NEXT dn
c% = ct% / 3
ct% = 0
c2% = CINT(c% / bol) - 5
PSET (j - n, i - m), c2%
NEXT j, i
END IF
CLOSE
LOCATE 30, 2: PRINT SPC(75);
END SUB

```

SUB zoom

' Seçilen çerçeve içindeki alanın, ekranın başka bir bölgesinde 5 kat büyütülmüş olarak çizilmesi.

```

DIM sakla%(32765)
yy1 = 420 - (y2 - y1) * 5
xx1 = 636 - (x2 - x1) * 5
h1 = xx1: h2 = yy1
r1 = 0: nn = 0
FOR r = y1 + 1 TO y2 - 1
xx1 = 636 - (x2 - x1) * 5
r1 = r1 + 1
FOR tt = 1 TO 5
yyy = yy1 + r1 + tt
p1 = 0
FOR P = x1 + 1 TO x2 - 1
p1 = p1 + 1
q = POINT(P, r)
FOR t = 1 TO 5
xxx = xx1 + p1 + t
nn = nn + 1
IF nn = 1 THEN xxx3 = xxx: yyy3 = yyy
sakla%(nn) = POINT(xxx, yyy)
PSET (xxx, yyy), q
NEXT t
xx1 = xx1 + 4
NEXT P
xx1 = 636 - (x2 - x1) * 5
NEXT tt
yy1 = yy1 + 4
NEXT r
DO
LOOP UNTIL INKEY$ = CHR$(27)
nn = 0
FOR yyy = yyy3 TO 416
FOR xxx = xxx3 TO 632
nn = nn + 1
PSET (xxx, yyy), sakla%(nn)
NEXT xxx, yyy
ERASE sakla%
END SUB

```