

T.C.
KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ

HİPERSPEKTRAL GÖRÜNTÜ SINIFLANDIRMA

MÜHENDİSLİK TASARIMI -2 PROJE ARA RAPORU

FERHAT GEYİK

170207049

DANIŞMAN: DOÇ. DR. ALP ERTÜRK

İçindekiler

1. GİRİŞ	3
1.1 PROJENİN AMACI.....	3
1.2 LİTERATÜR TARAMASI	3
1.3 HİPERSPEKTRAL GÖRÜNTÜ NEDİR?	4
1.4 HİPERSPEKTRAL GÖRÜNTÜLERDE SINIFLANDIRMA	6
2 K-NN YÖNTEMİ İLE SINIFLANDIRMA	6
3 ARA RAPOR SÜRESİNE KADAR YAPILAN ÇALIŞMALAR	8
4 ARA RAPOR DÖNEMİNDEN SONRA YAPILAN ÇALIŞMALAR.....	11
4.1 YAPAY SİNİR AĞLARI İLE SINIFLANDIRMA.....	11
4.2 DENEYSEL SONUÇLAR	13
5 KAYNAKÇA	15

1. GİRİŞ

Mühendislik tasarımı-2 dersi kapsamında görüntü işleme alanında bir takım çalışmalar yapılması hedeflenmiştir. Bu çalışmalarda kullanılacak olan veri hipersektral görüntü verisidir. Bu veri üzerinde hedef tespiti, kenar keskinleştirme, sınıflandırma, pan keskinleştirme gibi çalışmalar yapılabilmektedir. Bu proje kapsamında hiperspektral görüntülerde sınıflandırma üzerine durulacaktır.

1.1 PROJENİN AMACI

Bu projenin amacı, farklı algoritma çeşitlerini kullanarak hiperspektral görüntülerde sınıflandırma yapmaktır. Çalışma sonrasında kullanılan yöntemler karşılaştırılacaktır.

1.2 LİTERATÜR TARAMASI

Sınıflandırma; birbirleriyle ilişkisi olan, seçici özellik bakımından birbirleriyle benzerlik gösteren canlıları, nesneleri vs. belli bir kategoriye atamaktır. Birçok farklı bilimsel alanda sınıflandırma çalışmaları mevcuttur. Bu proje kapsamında uzaktan algılanarak elde edilen hiperspektral görüntülerde sınıflandırma çalışması yapılacaktır.

Hiperspektral görüntüler çok küçük dalga boyu aralıklarında çok fazla bant içerdiğinden dolayı her bir pikselin spektrumsal yansımasını detaylıca barındırırlar. Sınıflandırma çalışmalarında da bu detaylı bilgiler sayesinde, ince farklılıklar dahi ayırt edilebilerek, doğruluk oranı daha yüksek sınıflandırma çalışmaları yapılır.

Hiperspektral görüntüler üzerinde eğitilmiş sınıflandırma çalışmalarında bu güne kadar birçok farklı yöntem kullanılmıştır. Bunlardan bazıları; destek vektör makineleri [1], genetik algoritma [2], yapay sinir ağları [3], maksimum olabilirlik sınıflandırması [4] adlı yöntemlerdir.

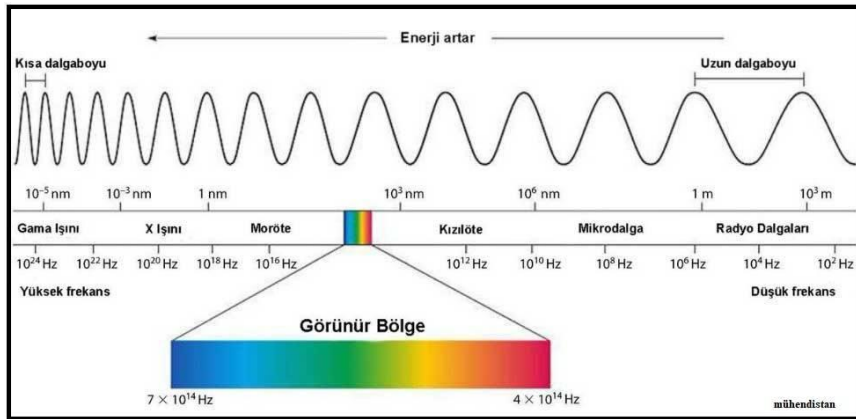
Son yıllarda büyük oranda gelişme gösteren hiperspektral sensörler yoğun oranda örnekleme sağlar. Benzer sınıflar arasında ayırım yapabilme konusunda kolaylık sağlar fakat bazen bu kadar yüksek kanal sayısı bazen gereksizdir. Bazı sınıflandırma çalışmalarında, sınıfları birbirinden ayırt edebilmek, daha az veri kullanarak mümkün olabilir. Daha az bant sayısı kullanarak sınıflandırma işlemi yapabilmek adına da literatürde özellik azaltma olarak geçen çalışmalar mevcuttur. Bunlar da kendi aralarında bazı gruplara ayrılmıştır. [5] Özellik çıkarımı yöntemlerini kullanarak işlem süresi azaltılır, depolama için daha düşük bellek kullanır ve verilerin daha iyi anlaşılmasını sağlar.

Yüksek boyutlu sınıflandırma çalışmalarında başarıyı arttırmak amacıyla geliştirilen yöntemlerden bayes kestirim teknikleri [6] ve karar ağaçları [7] yöntemleri de kullanılmıştır.

Bir başka eğitimli sınıflandırıcı yöntemi K-NN(K Nearest Neighbor) Türkçe’de K en yakın komşu kenar anlamını taşıyan yöntemdir [8]

1.3 HİPERSPEKTRAL GÖRÜNTÜ NEDİR?

Görüntüleme sistemlerinin birçoğu insan tarafından algılanabilen görülebilir dalga boyu aralığında görüntüleme yapmaktadır ve oluşan görüntü genellikle renk bilgisi olarak sunulur. Spektral görüntüleme, elektromanyetik spektrumda birden fazla bant kullanan görüntülemektir. Sıradan bir kamera, görünür spektrumdaki kırmızı, yeşil ve mavi üç dalga boyu bandı boyunca ışığı yakalarken, spektral görüntüleme bu üç bandın ötesine geçen çok çeşitli teknikleri kapsar. Spektral görüntüleme kızılötesi, görünür spektrum, ultraviyole, x-ışınları veya bunların bir kombinasyonunu kullanabilir.



Şekil 1. Elektromanyetik Spektrum [9]

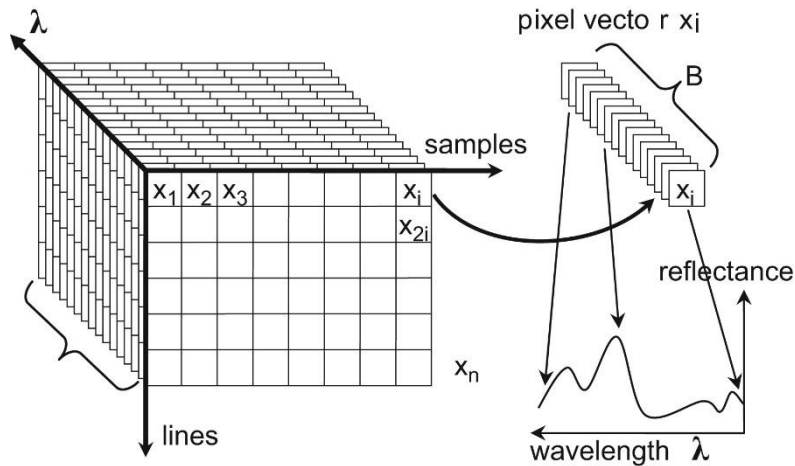
Şekil 1’de görüldüğü gibi optik bölge, 400nm-700nm aralığındaki mavi, yeşil, kırmızı renk bantlarından oluşan görülmür ışık bölgesi ve 700nm-1500nm dalga boyuna sahip ışınların oluşturduğu bölge yakın kızıl ötesi bölgesinden oluşmaktadır. 1500nm-3000nm dalga boyu aralığı kısa dalga boyu kızıl ötesi iken 3000nm-5000nm dalga boyu aralığında kalan kısım orta

dalga boyu kızılötesi olarak adlandırılmıştır. Optik bölgenin en solunda kalan kısım ise uzun dalga boyu kızılötesi olarak isimlendirilmiştir.

Görünür ve görünmeyen bantlarda aynı anda görüntü verilerinin alınmasını, görünür aralığın dışından aydınlatmayı veya belirli bir spektral aralığı yakalamak için optik filtrelerin kullanımını içerebilir. Bir görüntüdeki her piksel için yüzlerce dalga boyu bandı yakalamak da mümkündür.

Spektral çözünürlüklerine göre, algılanan görüntü pankromatik, multispektral ve hiperspektral olarak üç başlık altında toplanabilir. Pankromatik algılayıcılar algılanan cisim üzerinden yansıyan ışınların enerji seviyesine göre tek bir banttan oluşan görüntü oluştururlar. Bu algılayıcılar genellikle görünür bölge ile yakın kızılötesi bölge aralığında kalan dalga boylarında çalışırlar. Multispektral algılayıcılar ise genel olarak elektromanyetik spektrumda görünür bölge ile kısa dalga boyu kızıl ötesi bölgesi arasında çalışır ve 4 ile 7 arası bant içeren görüntüler oluştururlar. Multispektral görüntülerde her bir bantın genişliği 300nm-400nm arasındadır. Multispektral görüntüleme, değişken filtreler ve aydınlatma kullanılarak tipik olarak üç ila on beş arasında az sayıda spektral bantı yakalar. [10]

Hiperspektral görüntüleme, spektroskopi ve dijital fotoğrafçılığı birleştiren spektral görüntülemenin bir başka alt kategorisidir. Multispektral görüntüleme teknolojisinin gelişmişisi olarak da düşünülebilir. Uydu, uçak ya da farklı hava araçları üzerinde bulunan özel algılayıcılar tarafından elde edilen hiperspektral görüntüler 10nm-20nm dalga boyu aralığında değişen yüzlerce bant içerebilmektedir. Hiperspektral görüntülemeye kaydedilen spektrumlar, ince dalga boyu çözünürlüğüne sahiptir ve geniş bir dalga boyu aralığını kapsar. Hiperspektral görüntüleme aralıklı spektral bantları ölçen multispektral görüntülemenin aksine sürekli spektral bantları ölçer.



Şekil 2. Hiperspektral Görüntü Yapısı [11]

Her piksel için ayrıntılı bir ışık spektrumu kaydeden hiperspektral görüntüleme, farklı malzemelerin fiziksel doğası hakkında paha biçilmez bir bilgi kaynağı sağlayarak daha doğru bir sınıflandırma potansiyeline yol açar. Mühendisler astronomi, tarım, moleküler biyoloji,

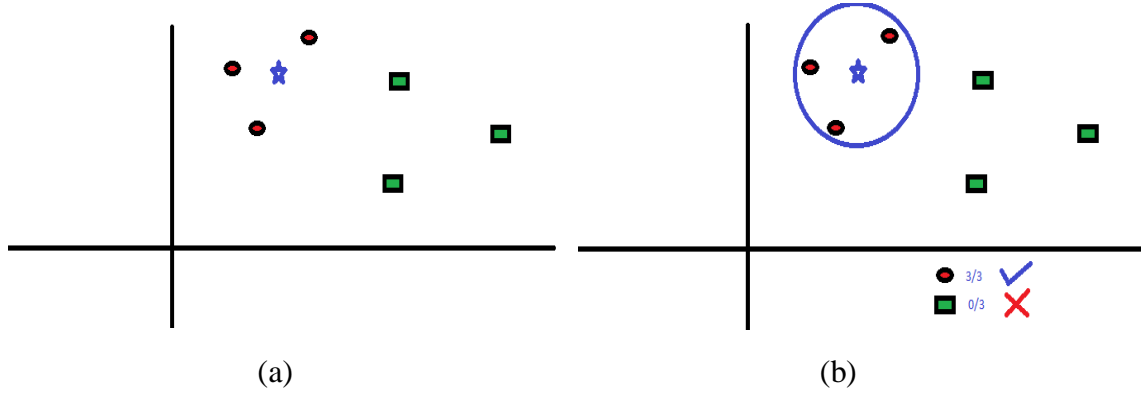
biyomedikal görüntüleme, yer bilimleri, fizik ve gözetim uygulamaları için hiperspektral sensörler ve işleme sistemleri kurarlar. Hiperspektral sensörler, elektromanyetik spektrumun büyük bir bölümünü kullanarak nesnelere bakar. Bazı nesneler elektromanyetik spektrumda benzersiz 'parmak izleri' bırakır. Spektral imzalar olarak bilinen bu 'parmak izleri' taranan bir nesneyi oluşturan malzemelerin tanımlanmasını sağlar. Örneğin, petrolün spektral imzası jeologların yeni petrol sahaları bulmasına yardımcı olur. [12]

1.4 HİPERSPEKTRAL GÖRÜNTÜLERDE SINIFLANDIRMA

Hiperspektral görüntülerde sınıflandırma yapılmasının amacı her piksel vektörüne, kendisine en yakın ilişkisi bulunan spektral imzayı taşıyan piksel vektörünün etiketini atabilmek, bu işlemi bütün pikseller için yapabilmektir. Sınıflandırma işlemi temelde ikiye ayrılır; eğitilmiş sınıflandırma ve eğitimsiz sınıflandırma. Bu projede eğitilmiş sınıflandırma üzerine durulup bu sınıflandırma çeşidi ile alakalı çalışmalar yapılacaktır. Eğitilmiş sınıflandırmada, sınıflandırma işleminden önce elimizde spektral imzaları bilinen ve buna göre de etiketler atanmış bir veri seti bulunmaktadır. Eğitilmiş sınıflandırma işleminde referans noktası bu spektral imzalar olacaktır. Daha önce konumu ve etiketi belirlenmiş bu veri setinden belli oranda rasgele örnekler alarak, sınıflandırma sürecinde hiperspektral verideki her pikseli bu veri setinden alınan, konumları bilinen örneklerle farklı algoritma yöntemleriyle işleme sokarak etiket atama işlemi yapılacaktır. Veri setinden rasgele seçilen etiketli piksellerin dışında kalan diğer etiketli pikseller ise test verisi olarak adlandırılır. İşlem bittikten sonra başarımlar oranı bu test verisi üzerinden hesaplanır.

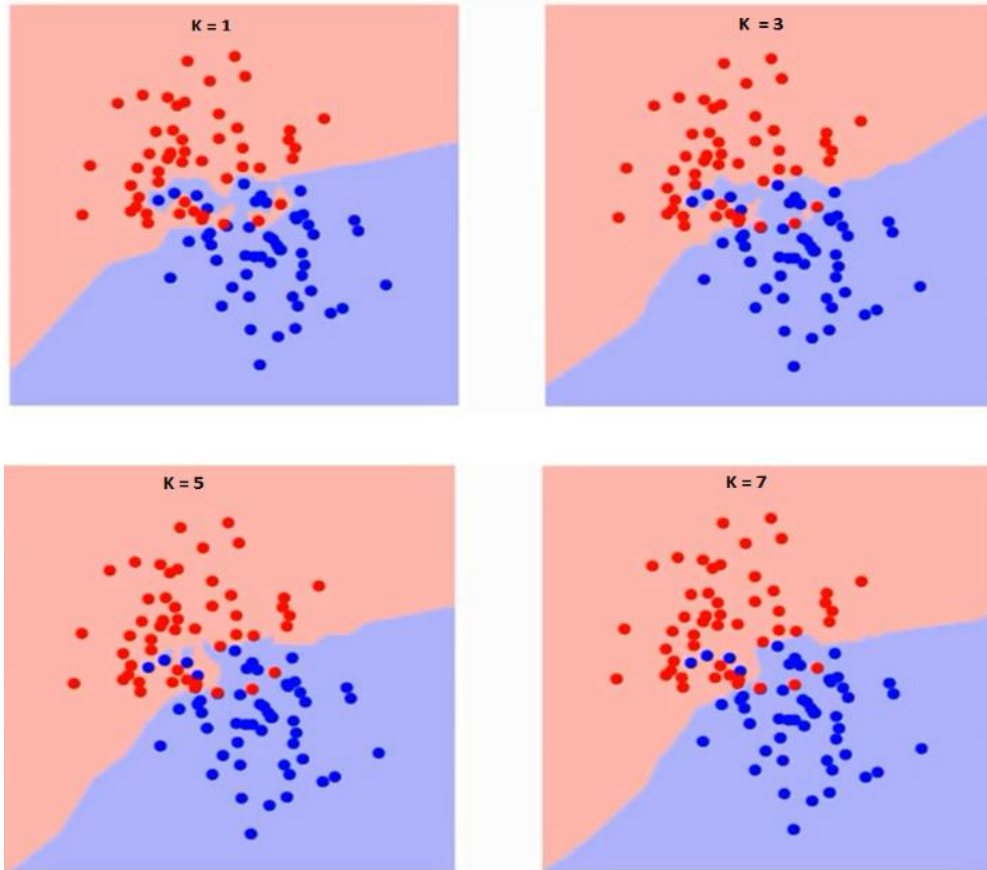
2 K-NN YÖNTEMİ İLE SINIFLANDIRMA

K-NN (K nearest neighbor) algoritması sınıflandırma problemlerinin çözümünde kullanılabilen, algoritması basit, uygulanması kolay, denetimli, çok eski bir makine öğrenmesi algoritmasıdır. K en yakın komşu olarak Türkçe'ye çevrilebilir. Burada bahsedilen K sayısı kullanıcı tarafından alınır ve genelde tek sayı olarak girilmesi tavsiye edilir. Algoritmanın temel mantığı, sınıflandırma işlemine tabii tutulacak olan verinin kendisine en yakın K adet veriden en çok hangisine yakın olduğu şeklindedir. K sayısı burada önem arz etmektedir K sayısı arttıkça bir yere kadar başarımlar oranı artmaktadır fakat K sayısını daha büyük seçmek de büyük veriler için oldukça büyük bir işlem yükü doğuracak, daha kısa işlem süresinde sonuç veren başka algoritmalar aranmasına yol açacaktır.



Şekil 3. (a) Sınıflandırılacak örnek mavi yıldız noktası, (b) K-NN algoritmasının mavi yıldız noktasını atadığı sınıf. [13]

Temelde şekil 3’de görüldüğü gibi çalışmaktadır. İşaretli konumu bilinen yıldız noktasını kendisine en yakın bulunan sınıf grubuna dahil etmektir amaç.



Şekil 4. Farklı K değerleri için sınıflandırma sonuçları [13]

Şekil 4’de görüldüğü üzere K sayısını arttırmak bize her zaman daha doğru sonucu verecek diye bir çıkarım elde etmek söz konusu değildir. Hem işlem yükü doğuracaktır bu durum hem de zaman zaman yanlış atama yapmaya müsait bir durum doğuracaktır. Asıl önemli olan veri seti üzerinden alınan eğitim setinin oranıdır. Bu eğitim seti oranı ne kadar yüksek alınırsa başarımlarımız yüzdesinde de o denli artış görmek mümkündür.

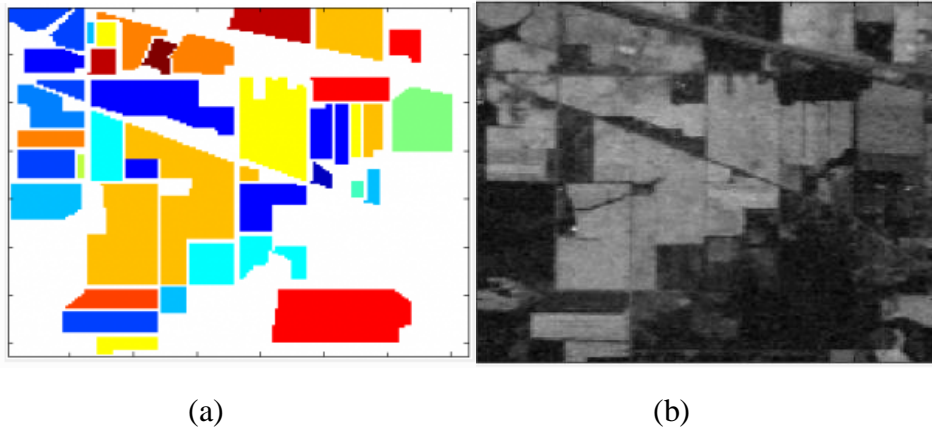
3 ARA RAPOR SÜRESİNE KADAR YAPILAN ÇALIŞMALAR

Hiperspektral görüntülerde sınıflandırma ile alakalı kaynak taraması yapıldı. K-NN algoritmasının çalışma mantığı kavrandı. K-NN algoritması ile sınıflandırma işlemi yapan kod yazılıp, farklı K değerleri ve veri setinden alınan farklı oranlarda eğitim setleri ile sınıflandırma yapıp, deneysel sonuç olarak başarımlar kaydedildi.

Eğitim ve test verisi oluşturma işlemi yer doğrusu (ground-truth) haritası üzerinden yapılır. Bu harita hiperspektral verinin daha önceden sınıfı bilinen piksellerinin işaretlenmiş olanıdır. Bu işaretli olan piksellerden belli bir oranda alınarak elde edilen veri seti eğitim seti olarak tanımlanır ve sınıflandırma işleminde bu eğitim setindeki yer vektörlerinin spektral imzaları referans alınarak sınıfı bilinmeyen diğer piksel vektörlerine sınıf ataması yapılır. Yer doğrusu haritasından eğitim seti olarak seçilmemiş olan diğer etiketli veriler ise test verisi olarak adlandırılır. Sınıflandırma işleminin sonunda başarımlar (overall accuracy) bu test verisi kullanılarak hesaplanır.

Başarımlar hesaplanırken, yer doğrusu haritası üzerindeki etiketli olan her pikselin etiketi ile sınıflandırma sonucunda elde edilen etiketlenmiş yeni haritanın aynı pikselin etiketi karşılaştırılır. Birebir doğru olarak atanmış etiket sayısının tüm etiket sayısına oranı başarımlar oranı olarak nitelendirilir.

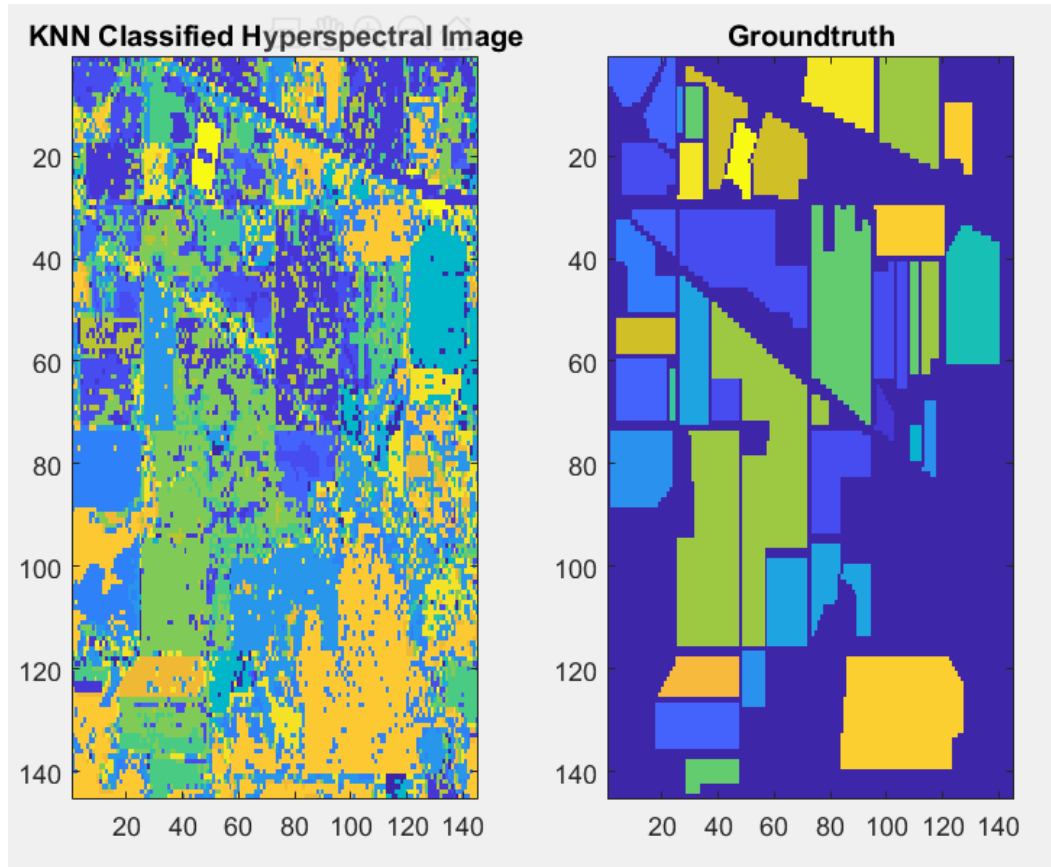
Deneysel çalışmalarda iki farklı veri seti kullanılmıştır. AVIRIS sensörü tarafından kuzey-batı Indiana'dan alınmış indian pines veri seti 145*145 boyutlarında 224 banttan oluşuyor. Bazı bantlar çıkarılarak 200 bantla düşürülmüştür bant sayısı [14]. Bu veri setinde toplamda 16 sınıf vardır.



Şekil 5. (a) indian pines veri seti yer doğrusu görüntüsü, (b) indian pines veri seti örnek bant görüntüsü [14]

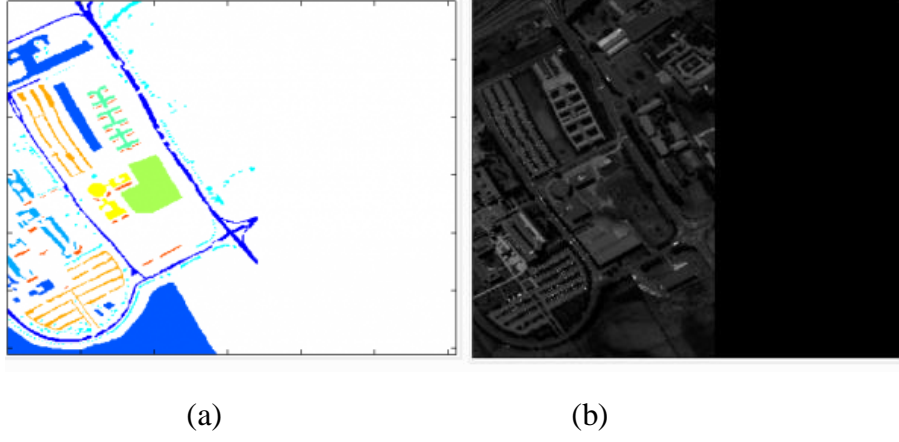
	K=1	K=3	K=5	K=9
Eğitim oranı %1	%38.64	%34.21	%34	%36.27
Eğitim oranı %2	%45.01	%44.26	%41.92	%40.52
Eğitim oranı %5	%50.04	%48	%47.26	%46.65
Eğitim oranı %10	%50.9055	%50.883	%50.4049	%50.5025

Tablo 1. indian pines veri seti üzerinden farklı K değerleri ve Eğitim oranları için başarımları(overall accuracy)



Şekil 6. K=3 ve %10 eğitim seti oranında K-NN sınıflandırma algoritmasıyla sınıflandırılmış indian pines verisi.

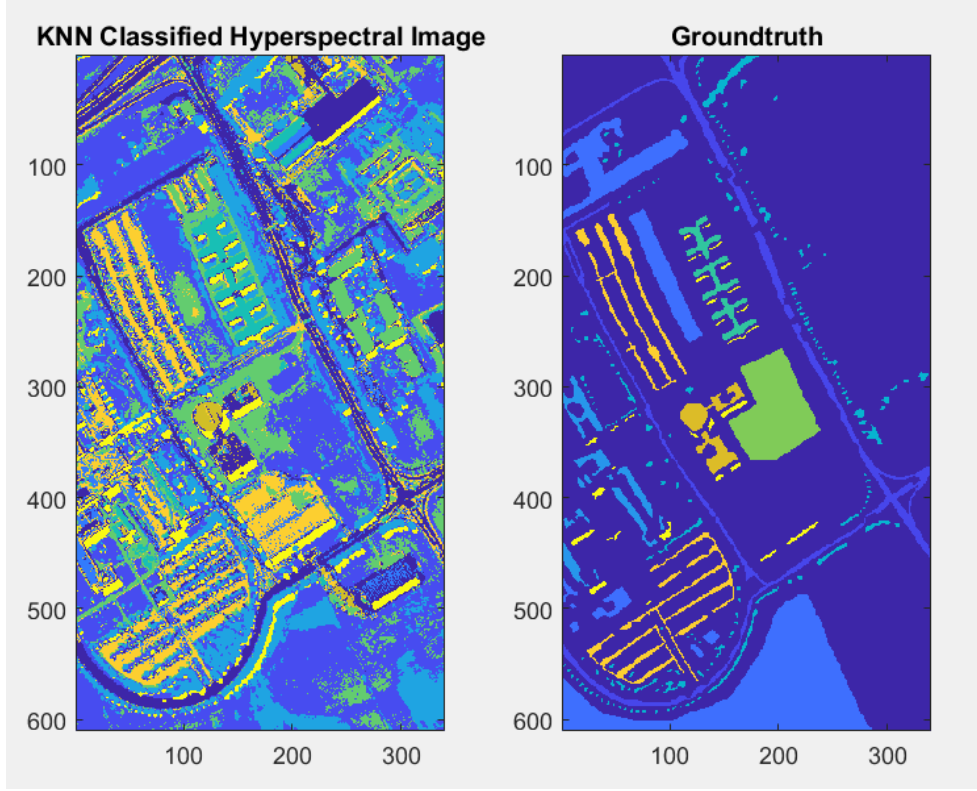
Çalışmada kullanılan bir diğer veri seti ROSIS sensörü ile Pavia Üniversitesi tarafından alınan pavia veri setidir. Bu hiperspektral görüntü 610*340 boyutlarında ve toplamda 103 banttan oluşur. Bu veri setinde toplamda 9 sınıf vardır.



Şekil 7. (a) Pavia Üniversitesi veri seti yer doğrusu görüntüsü, (b) Pavia Üniversitesi veri seti örnek band görüntüsü. [14]

	K=1	K=3	K=5	K=9
Eğitim oranı %1	%63.4959	%67.248	%68.2766	%68.3023
Eğitim oranı %2	%69.5063	%70.9136	%70.7313	%69.8289
Eğitim oranı %5	%70.1141	%70.6471	%70.8902	%71.0281
Eğitim oranı %10	%72.9895	%73.055	%73.048	%72.9241

Tablo 2. Pavia Üniversitesi veri seti üzerinden farklı K değerleri ve Eğitim oranları için başarımlar (overall accuracy)



Şekil 8. K=3 ve %10 eğitim seti oranında K-NN sınıflandırma algoritmasıyla sınıflandırılmış Pavia Üniversitesi verisi.

4 ARA RAPOR DÖNEMİNDEN SONRA YAPILAN ÇALIŞMALAR

4.1 YAPAY SİNİR AĞLARI İLE SINIFLANDIRMA

Yapay sinir ağları insanın fizyolojik sinir hücresi yapısından esinlenerek geliştirilmiş, sinir sisteminin çalışmasını taklit eden bir algoritmadır. Yapay sinir ağı da hücrelerden oluşur ve çalışma mantığı biyolojik hücrenin çalışmasını taklit eder. Yapay sinir ağları algoritmaları günümüzde pek çok farklı alanda kompleks problemlerin çözümünde kullanılmaktadır.

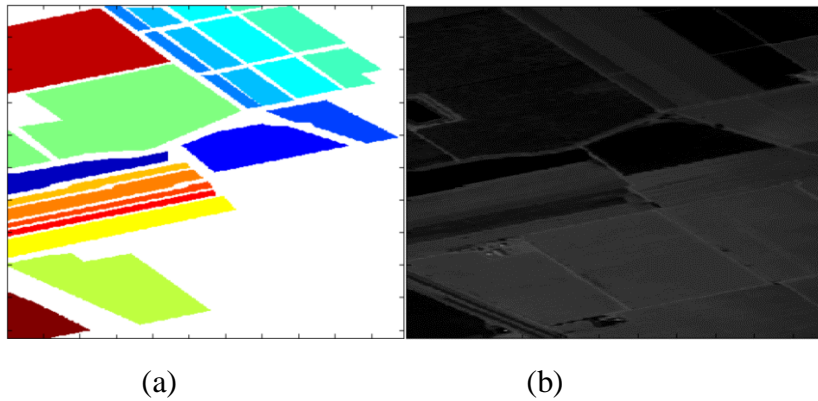
Yapay sinir ağlarında birçok farklı ağ topoloji modeli mevcuttur. Bu proje kapsamında basit tek katmanlı yapay sinir ağı(perseptron) kullanılmıştır. Bu modelde bütün girişler teker teker tüm hücrelere girdi olarak uygulanır ve bu girdilere her hücre farklı çıktı verir. Hiper spektral görüntülerde sınıflandırma probleminde bu çıktılar öklid mesafesi olup bizim aradığımız şey ise en kısa öklid mesafesine sahip hücrenin kim olduğunu bilmek olacaktır. Öğrenme aşaması devam ederken hücrelerin verdiği çıktıların doğru veya yanlış olduğuna bakılır. Destekleyici öğrenme olarak bilinen bu yöntemi kullanan model ise LVQ(Linear Vector Quantization) modelidir. Bu modellemede oluşturulan ilk hücrelere bize referans olarak verilen yer vektörü haritasından rasgele seçilmiş etiketini bildiğimiz konumların vektörünü ilk yol ağırlığı olarak atarız. Her farklı sınıftan belli bir oranda hücre oluşturup ilk yol ağırlıklarını atama işlemini

yaptıktan sonra öğrenme aşamasına geçilir. Öğrenme aşamasında yine etiketi bilinen vektörler teker teker bütün hücrelere girdi olarak uygulanır. Daha sonra dört farklı durum için yol ağırlıklarının güncellenmesi işlemi yapılır. Bu durumlardan birincisi en kısa mesafeye sahip ilk iki hücrenin de etiketinin doğru olması durumudur. Bu durumda birinci hücrenin yol ağırlığı kuvvetlendirilirken ikinci hücrenin yol ağırlığı zayıflatılır. İkinci durum ise en kısa mesafeye sahip iki hücrenin de etiketinin yanlış olması durumudur. Bu durumda yerel en iyi diye adlandırılan, aslında hücrelere giriş olarak uygulanan vektörün etiketine sahip en kısa öklid mesafesine sahip aynı etiketli hücre bulunur ve onun yol ağırlığı kuvvetlendirilirken genel en iyinin yol ağırlığı zayıflatılır. Üçüncü durum en kısa öklid mesafesine sahip ilk hücrenin etiketinin yanlış ikinci hücrenin etiketinin doğru olması durumudur. Bu durumda birinci hücrenin yol ağırlığı zayıflatılıp ikinci hücrenin yol ağırlığı kuvvetlendirilir. Son durum ise en kısa mesafeye sahip birinci hücrenin etiketinin doğru ikinci hücrenin etiketinin yanlış olması durumudur. Bu durumda yol ağırlıkları güncellenmez çünkü diğer bahsedilen üç durumda zaten ulaşılmak istenen nokta burasıdır.

Güncelleme işlemi sırasında yol ağırlıklarının kuvvetlendirilip zayıflatma işlemi; esasında eldeki çıktının öklid mesafesinin bir öğrenme katsayısıyla çarpılıp ilgili hücrenin vektörüne eklenmesi veya vektöründen skaler olarak çıkarılmasıdır. Öğrenme katsayısının değerinin ne olacağı önemlidir standart bir değeri yoktur. Bu proje kapsamında kullanılan değer; çıkan öklid mesafelerinin birkaçının sayısal olarak büyüklüğü göz önünde tutularak bir de aynı etikete sahip piksellerinin aynı bantlarının aralarındaki mesafeler karşılaştırılarak gözlemlene yoluyla belli bir değerde tutulmuştur. Sonuçta çıkan öklid mesafesini bu öğrenme katsayısıyla çarpıp yol ağırlıklarını güncelleştireceğimiz için daha büyük öğrenme katsayıları için oldukça yüksek hata oranları ekleyip çıkaracak bu da daha başarılı bir sınıflandırma sonucuna yakınsamaktan çok ıraksamak doğuracaktır.

Ara rapor öncesi kullanılan overal accuracy oranına ek olarak ara rapor dönemi sonrası çalışmalarda average accuracy değeri de hesaplanmıştır. Average accuracy değeri ise her sınıfın başarısını kendi içinde değerlendirerek farklı her sınıf için farklı bir başarı yüzdesi sonucu vermektedir.

Ara rapor öncesi kullanılan veri setlerine ek olarak kullanılan bir diğer veri seti; Salinas Valley, California bölgesine ait AVIRIS sensörü tarafından alınan salinas veri setidir. Bu hiperspektral görüntü 512*227 boyutlarında ve toplamda 224 banttan oluşur. Bu veri setinde toplamda 16 sınıf vardır. [14]



Şekil 9. (a) Salinas veri seti yer doğrusu görüntüsü, (b) Salinas veri seti örnek band görüntüsü [14]

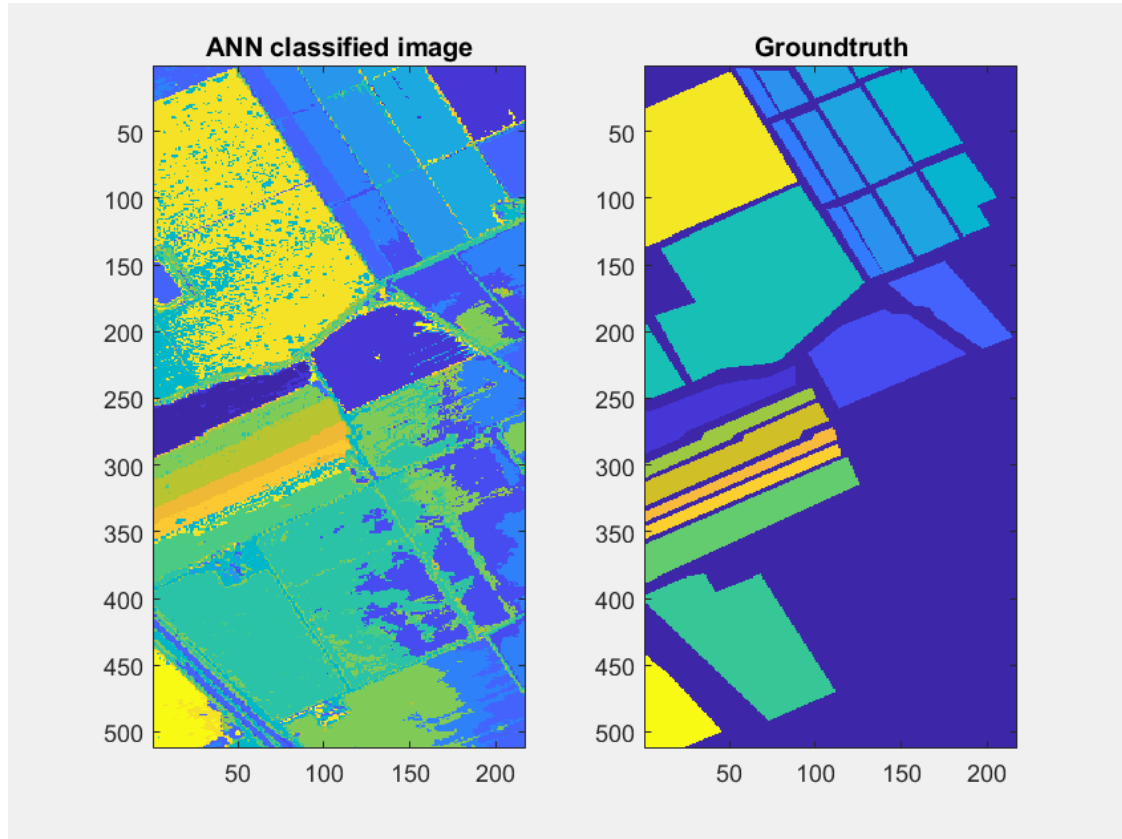
4.2 DENEYSEL SONUÇLAR

	Ratio2=0.1		Ratio2=0.3		Ratio2=0.5		Ratio2=0.7	
	OK=0.004 Ratio1=0.01	OK=0.003 Ratio1=0.02	OK=0.004 Ratio1=0.01	OK=0.003 Ratio1=0.02	OK=0.004 Ratio1=0.01	OK=0.003 Ratio1=0.02	OK=0.004 Ratio1=0.01	OK=0.003 Ratio1=0.02
Salinas	%70.54	%74.41	%71.07	%74.95	%68.97	%72.87	%68.13	%73.10
Indian pines	%36.57	%43.75	%38.62	%45.78	%37.97	%47.15	%35.75	%47.12
Pavia University	%65.85	%69.55	%62.09	%70.91	%65.27	%69.70	%65.57	%66.75

Tablo 3. Yapay sinir ağı kullanılarak yapılan sınıflandırmada farklı veri setleri üzerinden farklı parametre değerleri için başarımlar (overall accuracy)

Tabloda ratio2 olarak geçen parametre değeri eldeki veri setinden yüzde kaç oranında piksel sayısının eğitim aşamasına alınacağını belirler. Bu tabloda eğitim oranları olarak %10, %30, %50 ve %70 seçilmiştir. Ratio1 ise yine veri setinden alınan örneklerin yüzde kaç kadar hücre oluşturulacağını belirler. Bu tabloda oluşturulacak hücre sayısının oranları %1 ve %2 olarak seçilmiştir. Son olarak OK diye belirtilen parametre ise öğrenme katsayısının yerini tutmaktadır. Bu çalışmada 0.004 ve 0.003 özneler olarak belirlenmiştir.

Tablodan da anlaşılacağı üzere seçtiğimiz parametre değerleri her veri setinde aynı başarıya ulaştırmıyor. Farklı veri setleri için farklı parametre değerleri girilerek daha başarılı sonuçlar elde edilebilir. Bunun için daha kapsamlı bir araştırma gereklidir. Tablodan da anlaşılacağı üzere oluşturulan hücre sayısını iki katına çıkarmak genel olarak başarıda bir artışa neden olmuştur. Fakat eğitim oranını arttırmak başarıya sürekli pozitif yönde bir katkı sağlamamıştır.



Şekil 10. Ratio1=0.04, Ratio2=0.3 ve OK=0.003 seçilerek yapay sinir ağıyla sınıflandırılmış overall accuracy oranı=77.11 olan Salinas verisi.

#	SINIF İSMİ	ÖRNEK SAYISI	AVERAGE ACCURACY
1	Brocoli_green_weeds_1	2009	% 92.63
2	Brocoli_green_weeds_2	3726	% 95.41
3	Fallow	1976	% 66.70
4	Fallow_rough_plow	1394	% 98.92
5	Fallow_smooth	2678	% 81.47
6	Stubble	3959	% 96.51
7	Celery	3579	% 99.16
8	Grapes_untrained	11271	% 26.47
9	Soil_vinyard_develop	6203	% 98.03
10	Corn_senesced_green_weeds	3278	% 74.22
11	Lettuce_romaine_4wk	1068	% 91.19
12	Lettuce_romaine_5wk	1927	% 98.65
13	Lettuce_romaine_6wk	916	% 99.01
14	Lettuce_romaine_7wk	1070	% 84.67
15	Vinyard_untrained	7268	% 87.64
16	Vinyard_vertical_trellis	1807	% 84.28

Tablo 4. Şekil 10.'da belirtilen parametre değerleri kullanılarak yapay sinir ağıyla sınıflandırılmış Salinas veri setinin average accuracy oranları.

5 KAYNAKÇA

- [1] «F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with Support Vector Machines," IEEE Trans. Geosci. Remote Sens., vol. 42, no.».».
- [2] « D. G. Stavrakoudis, G. N. Galidaki, I. Z. Gitas, and J. B. Theocharis "A genetic fuzzy-rule-based classifier for land cover classification from hyperspectral imagery,"IEEE Trans. Geosci. Remote Sens., vol. 50, no. 1, pp.130–148, Jan. 2012.».
- [3] «F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," IEEE Trans. Geosci. Remote Sens., vol.48, no. 5, pp. 2271–2282, May 2010».
- [4] X. Jia, «Simplified maximum likelihood classification for hyperspectral data in cluster space,» *IEEE International Geoscience and Remote Sensing Symposium*, 2002.
- [5] M. Imani ve H. Ghassemian, «Feature extraction using partitioning of feature space for hyperspectral images classification,» *2014 Iranian Conference on Intelligent Systems (ICIS)*, 2014.
- [6] Q. & L. Jackson, «Adaptive bayesian contextual classification based on Markov random fields,» *IEEE Transactions on Geoscience and Remote Sensing*,, no. 40(11), p. 2454–2463, 2002.
- [7] H. d. O. M. D. & H. V. Bittencourt, «A binary decision tree classifier implementing logistic regression as a feature selection and classification method and its comparison with maximum likelihood,» *In Proceedings of IEEE International Geoscience Remote Sensing Symposium*, p. 1755–1758, 2007.
- [8] S. L. X. K. a. K. H. Weiwei Song, «HYPERSPETRAL IMAGE CLASSIFICATION BASED ON KNN SPARSE,» *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.
- [9] «<https://muhendistan.com/elektromanyetik-radyasyon-nedir/>,» [Çevrimiçi].
- [10] «https://en.wikipedia.org/wiki/Hyperspectral_imaging,» [Çevrimiçi].
- [11] «Segmentation and classification of hyperspectral images using watershed transformation».
- [12] «https://tr.wikipedia.org/wiki/Hiperspektral_görüntüleme,» [Çevrimiçi].
- [13] «<https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>,» [Çevrimiçi].
- [14] «http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes,» [Çevrimiçi].

K-NN sınıflandırma kodu

```
close all;

clear all;

ratio = input('What is the ratio of TrainData: '); %eğitim oranı

knn = input('number of K?: '); %k sayısı


HSI = load('salinas.mat'); %hiperspektral görüntü
HSI = HSI.salinas;
HSI_GT = load('salinas_gt.mat'); %Groundtruth
HSI_GT = HSI_GT.salinas_gt;


[spat1,spat2,spec]=size(HSI);
[row, col, val] = find(HSI_GT); %val= toplam etiktli piksel sayısı
Num_of_Classes = max(HSI_GT(:)); %kaç tane sınıf olduğu bulunur


t=0;
for n=1:1:Num_of_Classes %sınıf sayısı kadar döğüye girsin
    [row, col, val] = find(HSI_GT == n); %bir sınıfın satır sütun ve etiket bilgilerini bul
    [Num_of_Train , one] = size(val);
    a=round(Num_of_Train*ratio);
    b=randperm(a,a); % rasgele pikseller seçme
    for i=1:1:a
        label_location(i+t,1)=row(b(i)); %eğitim için alınan piksellerin satır sütun ve etiket bilgisini ayrı bir
        matriste tut
        label_location(i+t,2)=col(b(i));
        label_location(i+t,3)=HSI_GT(row(b(i)),col(b(i)));
    end
    t=t+a;
end
```



```

% K-NN sınıflandırma

distance=0;

counter=size(label_location,1);

neighbors=zeros(1,knn);

tagged=zeros(spat1,spat2); %sınıflandırma sonucunu gösterecek matris

for x=1:spat1
    for y=1:spat2
        for z=1:counter %eldeki bütün eğitim örneklerini dolaşacak
            for band=1:spec
                distance = distance + (HSI(x,y,band)-HSI(label_location(z,1),label_location(z,2),band))^2; %öklid
mesafesi
            end
            dist(z,1)=sqrt(distance);
            distance=0;
        end
        [v , index]=sort(dist(:,1)); %en düşük mesafeye sahip olana göre sırala
        for k=1:knn
            neighbors(k)=label_location(index(k),3); %K sayısı kadar en kısa mesafeye sahip örnek etiket değerini al
        end
        tagged(x,y)=mode(neighbors); % en çok tekrarlayanı seç
    end
end

figure;

subplot(1,2,1); imagesc(tagged); title('KNN Classified Hyperspectral Image');

subplot(1,2,2); imagesc(HSI_GT); title('Groundtruth');

```

```

% başarıım oranı overall accuracy

true=0;
false=0;
for x=1:spat1
    for y=1:spat2
        if(HSI_GT(x,y)~=0)
            if(tagged(x,y)==HSI_GT(x,y))
                true = true+1;
            else
                false = false+1;
            end
        end
    end
end

success_rate= true*100 / (true + false);
print=['overall accuracy of KNN Classification = ',num2str(success_rate)];
disp(print)

```

Yapay Sinir Ağı sınıflandırma kodu

```

close all;

clear all;

HSI = load('salinas.mat');      %hiperspektral görüntü

HSI = HSI.salinas;

HSI_GT = load('salinas_gt.mat'); %Groundtruth

HSI_GT = HSI_GT.salinas_gt;


ratio1 = 0.04; % etiketi bilinen piksellerin yüzde kaç kadar hücre oluşturulsun

ratio2= 0.3; % yol ağırlıklarının güncellenme işlemindeki iterasyon sayısına denk gelir

OK = 0.003; % öğrenme katsayısı

[spat1,spat2,spec]=size(HSI);

[row, col, val] = find(HSI_GT);

Num_of_Classes = max(HSI_GT(:)); %kaç farklı sınıf var

```

```

t=0;
% her etiketten rasgele noktalar seçme
for n=1:1:Num_of_Classes
    [row, col, val] = find(HSI_GT == n);
    [Num_of_sample , one] = size(val);
    a=round(Num_of_sample*ratio1);
    if(a<8) %her sınıftan en az 8 hücre oluşturulsun.
        a=8;
    end
    b=randperm(a,a);% rasgele seçim
    for i=1:1:a
        noron_points(i+t,1)=row(b(i)); % seçilen piksellerin satır sütun ve etiket bilgisini ayrı bir matrsite tut
        noron_points(i+t,2)=col(b(i));
        noron_points(i+t,3)=HSI_GT(row(b(i)),col(b(i)));
    end
    t=t+a;
end
n=size(noron_points,1); % n= hücre sayısı

norons=zeros(n,1,spec); % örnek sayısı kadar yeni hücre oluştur.
norons_labels=zeros(1,n); % hücre etieketleri için bir dizi daha oluştur.

for a=1:1:n    %hücrelere ilk yol ağırlıklarını atama
    norons(a,1,1:end)=HSI(noron_points(a,1),noron_points(a,2),1:end);
    norons_labels(1,a)=noron_points(a,3);
end

```

```

t=0;
% eğitim için her etiketten rasgele noktalar seçme
% hücre seçimi işleminin aynısı sadece oran farklı
for m=1:1:Num_of_Classes
    [row, col, val] = find(HSI_GT == m);
    [Num_of_sample , one] = size(val);
    k=round(Num_of_sample*ratio2);
    p=randperm(k,k);
    for r=1:1:k
        teach_points(r+t,1)=row(p(r));
        teach_points(r+t,2)=col(p(r));
        teach_points(r+t,3)=HSI_GT(row(p(r)),col(p(r)));
    end
    t=t+k;
end

counter=size(teach_points,1);

for i=1:1:counter %% eğitime girecek her örnek için

    for j=1:1:n %% bütün hücrelerle karşılaştır
        distance=0;
        for band=1:1:spec
            distance = distance + (HSI(teach_points(i,1),teach_points(i,2),band)-norons(j,1,band))^2;
        end
        %öklid mesafesi
        dist(j,1)=sqrt(distance);
    end
end

```

```
[v , index]=sort(dist(:,1)); % en küçük oklid mesafesine göre sıralama yapılır
```

```
E = OK*dist(index(1),1); % en düşük oklid mesafesini hata sayısı olarak tanımlayıp direkt öğrenme katsayısıyla çarpıp bir değişkene atadım
```

```
%LVQ-X yöntemine göre yol ağırlıklarının güncellenme işlemi
```

```
while(E~=0) % en düşük mesafenin 0 gelmesi olasılığına karşı E değerinin 0 olması durumunda döngüye girmemesi için.
```

```
if(((teach_points(i,3))== (norons_labels(1,index(1)))) && ((teach_points(i,3))== (norons_labels(1,index(2))))) %% en düşük mesafeye sahip ilk iki hücrenin de etiketi doğru mu ?
```

```
for k=1:1:spec
```

```
norons(index(1),1,k) = norons(index(1),1,k)+ E; %% 1. hücrenin yol ağırlığını kuvvetlendir
```

```
norons(index(2),1,k) = norons(index(2),1,k)- E; %% 2. hücrenin yol ağırlığını zayıflat.
```

```
end
```

```
elseif(((teach_points(i,3)) ~= (norons_labels(1,index(1)))) && ((teach_points(i,3)) ~= (norons_labels(1,index(2))))) %% en düşük mesafeye sahip ilk iki hücrenin de etiketi yanlış mı ?
```

```
for m=3:1:n %% en düşük mesafeye sahip yerel en iyiyi bul
```

```
if((teach_points(i,3))== (norons_labels(1,index(m))))
```

```
for o=1:1:spec
```

```
norons(index(m),1,o) = norons(index(m),1,o)+ E; %% yerel en iyinin yol ağırlığını kuvvetlendir.
```

```
norons(index(1),1,o) = norons(index(1),1,o)- E; %% genel en iyinin yol ağırlığını zayıflat.
```

```
end
```

```
break
```

```
end
```

```

end

elseif(((teach_points(i,3)) ~= (norons_labels(1,index(1)))) && ((teach_points(i,3)) ==
(norons_labels(1,index(2))))) %% 1. hücrenin etiketi yanlış 2. hücrenin etiketi doğru ise

    for s=1:1:spec
        norons(index(1),1,s) = norons(index(1),1,s) -E; %% 1. hücrenin yol ağırlığını zayıflat.
        norons(index(2),1,s) = norons(index(2),1,s) +E; %% 2. hücrenin yol ağırlığını kuvvlendir.
    end
else
end
E=0; %% 0 olmayan herhangi bir E değeri için döngünün bir kez dönmesi gerekiyor.
end
end
distance=0;
tagged=zeros(spat1,spat2); %yeni etiket değerleri için boş bir matris oluştuduk.
for x=1:1:spat1
    for y=1:1:spat2
        for z=1:1:n %% bütün hücrelere sor
            for band=1:1:spec
                distance = distance + (HSI(x,y,band)-norons(z,1,band))^2;
            end
            dist(z,1)=sqrt(distance);
            distance=0;
        end
        [v , index] = sort(dist(:,1));
        tagged(x,y)= norons_labels(1,index(1));
    end
end
end

```

```

figure;
subplot(1,2,1); imagesc(tagged); title('ANN classified image');
subplot(1,2,2); imagesc(HSI_GT);title('Groundtruth');


true=0;
false=0;
for x=1:spat1
    for y=1:spat2
        if(HSI_GT(x,y)~=0)
            if(tagged(x,y)==HSI_GT(x,y))
                true = true+1;
            else
                false = false+1;
            end
        end
    end
end

success_rate= true*100 / (true + false);
print=['overall accuravy of ANN classification = %',num2str(success_rate)];
disp(print)


average_accuracy=zeros(Num_of_Classes,1);


for n=1:1:Num_of_Classes
    [row, col, val] = find(HSI_GT == n);
    [num_of_label , one] = size(val);

    count=0;

```

```
for i=1:1:num_of_label
    if(tagged(row(i),col(i))==HSI_GT(row(i),col(i)))
        count=count+1;
    end
end
average_accuracy(n,1)=count*100/num_of_label;
end
```