

KOCAELİ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

**2019 IEEE Data Fusion Yarışmasında Kullanılan Veri Setleri Üzerinden Nokta Bulutu
Şeklindeki 3B Yapıların Sınıflandırılması**

LİSANS TEZİ

SAMET ARSLANTÜRK

Bölümü: Elektronik ve Haberleşme Mühendisliği

Danışman: Arş. Gör. Dr. Ali Can KARACA

KOCAELİ, 2020

ÖNSÖZ VE TEŞEKKÜRLER

Bu tez kapsamında günümüzde gitgide popülerleşen makine öğrenmesi ile sınıflandırma işlemi üzerine çalışmalar yapılmıştır. Bu amaçla Matlab programı kullanılarak, IEEE veri setleri kullanılarak belli analizler yapılarak, analiz sonuçları iyileştirilmeye çalışılmıştır.

Kullanılmış olan veri setlerini bize sağlamış olduğu için Johns Hopkins Üniversitesi Uygulamalı Fizik Laboratuvarı-IARPA'ya ve Veri Füzyon Yarışması düzenleyen IEEE GRSS Görüntü Analizi ve Veri Füzyon Teknik Komitesine teşekkürlerimi sunarım.

Ayrıca bu tez kapsamında benden dualarını ve maddi desteklerini hiç eksik etmeyen annem, babam ve amcam Ali ARSLANTÜRK'e, uygun koşullarda tezimi yapmama olanak sağlayan arkadaşım Mustafa AYDOĞDU ve her türlü koşulda bilgilerini son safhasına kadar bana aktarmaya çalışan ve her türlü desteği veren çok değerli danışman hocam Arş. Gör. Ali Can KARACA'ya minnetlerimi sunar, teşekkürü borç bilirim.

Bu tez kapsamında elde edilen analiz sonuçlarının bir nebze de olsun bilime katkı sağlayarak, ileride çalışma yapacak olan bilim arkadaşlarıma ışık tutması dileğiyle.

Aralık 2019, Kocaeli

Samet ARSLANTÜRK

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜRLER.....	i
İÇİNDEKİLER.....	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ	v
SİMGELER DİZİNİ VE KISALTMALAR.....	vi
ÖZET.....	vii
İNGİLİZCE ÖZET	viii
1. GİRİŞ	1
2. LİDAR.....	2
2.1 Lidar Nasıl Çalışır?.....	7
2.2 Nokta Bulutu.....	7
2.3 LAS Dataset.....	7
2.4 Lidar Işıklılık (Intensity) Değeri.....	8
2.5 Lidar Nokta Bulutu Sınıflandırması	9
3. LİDAR VERİLERİNİN GÖRSELLEŞTİRİLMESİ	11
3.1 Işıklılık (Intensity) Değerinin Görselleştirilmesi.....	12
4. MAKİNE ÖĞRENMESİNE GİRİŞ	14
4.1 Terminoloji	15
4.2 Öğrenme Yaklaşımları.....	15
4.2.1 Gözetimli Öğrenme (Supervised Learning).....	15
4.2.1.1 Sınıflandırma (Classification)	16
4.2.1.2 Regresyon (Regression)	18
4.2.2 Gözetimsiz Öğrenme (Unsupervised Learning)	19
4.2.2.1 Kümeleme (Clustering)	20
4.2.2.2 Boyut Azaltımı (Dimensionality Reduction)	21
4.2.2.3 Birliktelik Kuralı (Association Rule Mining)	21
4.2.3 Takviyeli Öğrenme (Reinforcement Learning).....	22
4.3 Makine Öğrenmesi Adımları	22
4.3.1 Öğrenme Süreci	22

4.3.2 Özellik Çıkartımı.....	22
4.3.3 Model Oluşturma	23
4.3.4 Değerlendirme-Test	23
4.3.4.1 Değerlendirme-Test Sürecinde Kullanılan Parametreler	23
4.3.5 İyileştirme	24
4.3.6 Tahmin	24
5. KULLANILAN SINIFLANDIRMA YÖNTEMLERİ	25
5.1 Destek Vektör Makineleri (SVM)	25
5.2 K-En Yakın Komşuluk (KNN, K-Nearest Neighbor)	28
6. DESTEK VEKTÖR MAKİNELERİ İLE LİDAR GÖRÜNTÜLERİNİN SINIFANDIRILMASI	30
7. FARKLI SINIFLARDA ÖZNİTELİK SAYISINDAKİ DEĞİŞİMİN SINIFLANDIRMAYA ETKİSİ.....	34
8. SaCR YÖNTEMİ VE PARAMETRELERİNİN LİDAR DATASETE GÖRE KONFIGURE EDİLMESİ.....	37
9. SaCR, SVM VE KNN DATASETLERİNİN EŞİT ŞARTLAR ALTINDA FARKLI DATASETLERDE SINIFLANDIRMA TEST SONUÇLARI.....	42
SONUÇLAR VE ÖNERİLER	49
KAYNAKLAR.....	51
ÖZGEÇMİŞ	52

ŞEKİLLER DİZİNİ

Şekil 2.1: Lidar taraması [2].....	2
Şekil 2.2: Otonom araç lidar taraması [3]	3
Şekil 2.3: Lidar lazer dönüş sayısı [4].....	4
Şekil 2.4: LAS dataset [5]	8
Şekil 2.5: Işıklılık değerinin görselleştirilmesi [6].....	9
Şekil 3.1:Lidarnokta bulutu veri setlerinin X, Y, Z koordinatlarına göre görselleştirilmesi ...	11
Şekil 3.2:Lidarverilerinin X, Y, Z koordinatları ve ışıklılık değerine göre görselleştirilmesi .	12
Şekil 3.3:Işıklılıkdeğerinin görselleştirilmesi	13
Şekil 4.1: Eğitim verileri [9]	16
Şekil 4.2: Gözetimsiz öğrenme yöntemi [10].....	19
Şekil 5.1: SVM genel yapısı [13]	26
Şekil 5.2: SVM sınıflandırma çeşitleri [14]	27
Şekil 5.3: Çekirdek hilesi ile 2 boyutlu grafiği 3 boyutlu hale getirmek [15].....	27
Şekil 5.4: KNN örnek çizimi [15]	28
Şekil 5.5: KNN çalışma adımları [16].....	29
Şekil 6.1: 4 özellik için Classification Learner işlemi sonucu	30
Şekil 6.2: 3 sınıf etiketi için elde edilen sonuçlar	31
Şekil 6.3: 4 özellik için elde edilen sonuçlar.....	32
Şekil 6.4: 6 özellik için Classification Learner işlemi sonucu	33
Şekil 6.5: 6 özellik için elde edilen sonuçlar.....	33
Şekil 9.4: JAX166 veri seti için elde edilen confusion matrix sonucu	44
Şekil9.1: JAX166 veri setinin SVM görselleştirme sonuçları	45
Şekil 9.2: JAX166 veri setinin KNN görselleştirme sonuçları	45
Şekil 9.3: JAX166 veri setinin SaCR görselleştirme sonuçları.....	46
Şekil 9.5: SVM, KNN ve SaCR test başarımları sonuçları ortalaması bar grafiği.....	48

TABLÖLAR DİZİNİ

Tablo 2.1: Lidar için depolanan özellikler ve açıklamaları [4]	6
Tablo 2.2: Lidar verisi sınıflandırma sayısal kodları [6].....	10
Tablo 8.1: $\Lambda=0.01$, $\gamma=1$ iken edilen c değerleri	38
Tablo 8.2: $C=2$, $\gamma=1$ iken elde edilen λ değerleri	39
Tablo 8.3: $C=2$, $\lambda=10^{-7}$ iken elde edilen γ değerleri	40
Tablo 8.4: $\Lambda=10^{-7}$, $\gamma=0.95$ iken elde edilen c değeri	41
Tablo 9.1: JAX166 veri seti için elde edilen sonuçlar	43
Tablo 9.2: OMA198 veri seti için elde edilen sonuçlar	46
Tablo 9.3. JAX280 veri seti için elde edilen sonuçlar	47

SİMGELER DİZİNİ VE KISALTMALAR

ASPRS	: American Society for Photogrammetry and Remote Sensing
GPS	: Global Positioning System
GPU	: Graphics Processing Unit
IEEE	: The Institute of Electrical and Electronics Engineers
IMU	: Inertial Measurement Unit
JAX_166_CLS	: Jacksonville/Floridada/USA için 166. Tile'dan alınmış CLS
JAX_166_PC3	: Jacksonville/Floridada/USA için 166. Tile'dan alınmış PC3
JAX_198_CLS	: Jacksonville/Floridada/USA için 198. Tile'dan alınmış CLS
JAX_198_PC3	: Jacksonville/Floridada/USA için 198. Tile'dan alınmış PC3
KNN	: K-Nearest Neighbors
LİDAR	: Light Dedection and Ranging
MATLAB	: Matrix Laboratory
OMA_198_CLS	: Omaha/Nebraska/USA için 198. Tile'dan alınmış CLS
OMA_198_PC3	: Omaha/Nebraska/USA için 198. Tile'dan alınmış PC3
PCA	: Principal Component Analysis
RGB	: Red Green Blue
SaCR	: Spatial-Aware Collaborative Representation
JSaCR	: Joint Spatial-Aware Collaborative Representation
SVM	: Support Vector Machines

2019 IEEE DATA FUSION YARIŞMASINDA KULLANILAN VERİ SETLERİ ÜZERİNDEN NOKTA BULUTU ŞEKLİNDE 3B YAPILARIN SINIFLANDIRILMASI

Samet ARSLANTÜRK

Anahtar Kelimeler: Makine Öğrenmesi, SVM, KNN, SaCR, Sınıflandırma, 3B Nokta Bulutu, IEEE Data Fusion Yarışması.

ÖZET

Günümüz de makine öğrenmesi uygulamaları gitgide popülerleşmeye ve verinin olduğu her yerde kullanılmaya başlamıştır. Makine öğrenmesi ile beraber yapay zeka ile ilgili gelişmelerde hızlı bir şekilde gerçekleşmektedir.

Yapılan bu tez kapsamında IEEE Data Fusion yarışmasında kullanılmış olan 3 boyutlu lazer nokta kümelerinin sınıflandırma çalışmaları yapılmıştır. Konu ile alakalı çalışmalar Matlab yazılımı ortamında gerçekleştirilmiştir.

Sınıflandırma için SVM, KNN ve SaCR yöntemleri kullanılmış. Değişken parametreler sonucunda elde edilen başarımlar oranları analiz edilerek, başarımlar oranları arttırılmıştır.

CLASSIFICATION OF 3D STRUCTURES AS POINT CLOUD ON DATA SETS USED IN THE 2019 IEEE DATA FUSION COMPETITION

Samet ARSLANTURK

Keywords: Machine Learning, SVM, KNN, SaCR, Classification, 3D Point Cloud, IEEE Data Fusion Competition.

ABSTRACT

Today, machine learning applications are becoming more and more popular and used everywhere. The development of artificial intelligence is realized rapidly with machine learning.

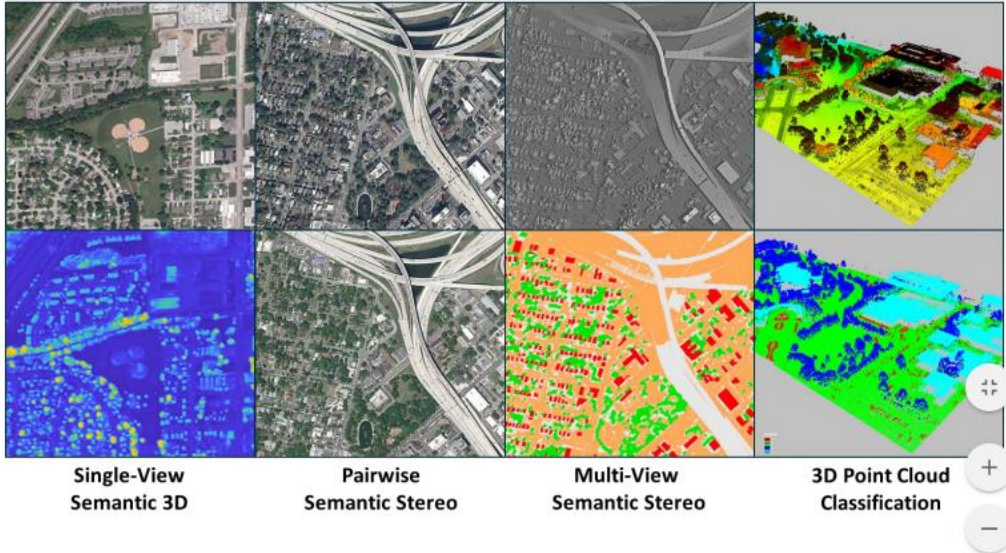
Within the scope of this thesis, classification studies of 3D laser point clusters used in IEEE Data Fusion competition have been made. The related studies were carried out in Matlab software environment.

SVM, KNN and SaCR methods were used as the classification method. The performance rates obtained as a result of the variable parameters were analyzed and the performance rates were increased.

1. GİRİŞ

Bu tez kapsamında lidar verilerinden alınan nokta bulutları ile yer şekillerinin sınıflandırılması üzerine çalışmalar yapılmıştır. Kullanılan veri setleri, Mart 2019 IEEE Data Fusion Yarışması'nda kullanılmıştır. Bu verilere internet üzerinden IEEE Data Fusion Yarışması'na kayıt yapılarak onay maili alındıktan sonra gelen linkler doğrultusunda uTorrent sitesi üzerinden indirilmiştir.

Data Fusion yarışması genel olarak 4 Track'ten oluşmaktadır ve bu tezde yarışmanın 4. Track'i olan 3D Point Cloud Classification veri setleri üzerinden gerçekleştirilmiştir. Data Fusion yarışması kategorilerine Şekil 1.1'den ulaşılabilir.



Şekil 1.1: IEEE Data Fusion yarışması kategorileri [1]

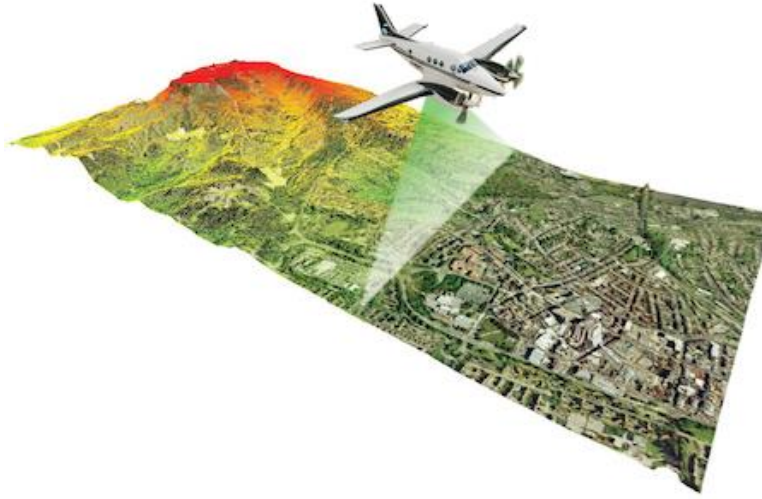
İndirilen dosyalar .txt formatında indirilmiştir ve track dosyalarında toplam 5 adet sütun ve sınıflandırma dosyalarında ise 1 adet sütun bulunmaktadır.

Yapılan işlemlerde öncelikli olarak Matlab programı üzerinden görselleştirilmeler yapılmaktadır. Ardından bu veriler Matlab programında SVM, KNN ve SaCR yöntemleri ile sınıflandırılmıştır.

2. LİDAR

Lidar dünya üzerinde sıkça kullanılan optik bir uzaktan algılama sistemidir. Lidar temel olarak ölçme işlemi için lazer veya ışık kullanan bir uzaktan algılama yöntemidir.

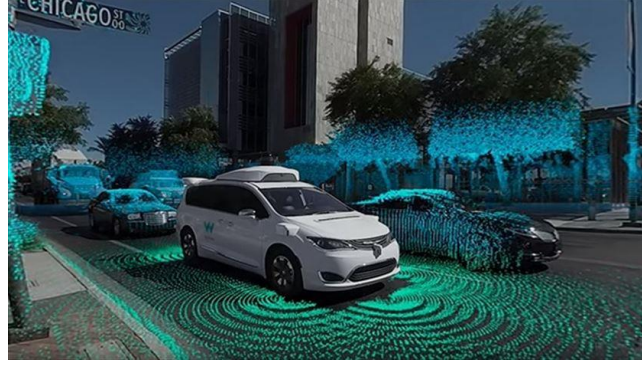
Açılımı “Light Dedection and Ranging” olan lidar sistemi özetle bir sinyal gönderildiği ve geri beklendiği anlamına gelmektedir. Yayılan sinyalin sensöre ne kadar sürede vardığı üzerinden işlem yapılarak konumlandırma işlemi yapılmaktadır. Bu işlemler sonucunda milyonlarca noktadan oluşan nokta bulutları oluşmaktadır. Lidar taraması ile alakalı görsele Şekil 2.1’den ulaşılabilir.



Şekil 2.1: Lidar taraması [2]

3 boyutlu lazer tarama tekniğiyle çalışan lidar 1960’lı yıllarda havadan deniz altılarının tespiti için geliştirilmiş olup, 1970’li yıllarda kullanılmaya başlanmıştır. Daha sonraki yıllarda ise hem havadan hem de karadan olmak üzere lidar tekniğini kullanan sensörlerin kullanım alanı ve çeşitlerinde artış meydana gelmiştir.

Harita sektöründe, karmaşık cisim veya bölgelerin, ölçülmesi imkansız yerlerin kurtarıcı yöntemi olmasına rağmen farklı sektörlerde de sık sık kullanılmaktadır. Bunların yanı sıra günümüzde git gide popülerleşen otonom araçlarda da lidar teknolojisi kullanılmaya başlanmıştır. Otonom araçlarda kullanılan lidar teknolojisi ile alakalı görsele Şekil 2.2’den ulaşılabilir.



Şekil 2.2: Otonom araç lidar taraması [3]

Ayrıca bir LİDAR sisteminin ana donanım bileşenleri şunlardır:

- 1-)Toplama aracı (uçak, helikopter, araç ve tripod),
- 2-) Lazer tarayıcı sistemi,
- 3-) GPS (Global Konumlandırma Sistemi),
- 4-)INS (Ataletsel Navigasyon Sistemi) içerir. Bir INS sistemi, LİDAR sisteminin yuvarlanma, adım ve yönünü ölçer.

Havadan tarama yapan lidar sistemleri, lidar sensörü, IMU ve dahili bilgisayardan oluşurken, karadan tarama yapan sistemlerde IMU bulunmaz. Bunların yanında GPS ve INS gibi birçok teknolojiye yardımcı olarak kullanılır. Her iki tür lidar sistemlerinde lazer kaynağından çıkan ışınlar sürekli olmayıp periyodik olarak yansıtıcıya çarpıp tarama yapılacak olan alana yönlendirilir. Böylece hava aracından veya karada tripot üzerinden tarama yapılacak alanın saniyede 20.000 ila 150.000 noktasının taraması gerçekleştirilir.

Bir birimlik sinyal herhangi bir zemine ulaştığı vakit o yüzeyden tekrar yansıtılarak önce yansıtıcıya sonra da alıcı sensöre ulaşır. Bu esnada sinyalin gidişi ile dönüşü arasındaki zaman farkı dâhili bilgisayar tarafından ölçülür. Bu zaman farkı ışık hızı ile çarpılıp ikiye bölündüğünde sensör ile görüntüsü alınan obje arasındaki uzaklığı verir. Bu işlemler sırasında havadan taramada GPS anlık olarak hava aracının coğrafi koordinatlarını alırken, IMU ise platformunun dikey eksene göre konum bilgisini alarak bu bilgiler lazer sensöründen alınan bilgilerle dahili bilgisayarda eşleştirilir.

Karada ise yine GPS coğrafi koordinatlar belirlenirken, sistem hali hazırda tripot üzerinde olduğundan dikey eksene göre konumu sabittir ve dâhili bilgisayar yardımı ile 360 derece

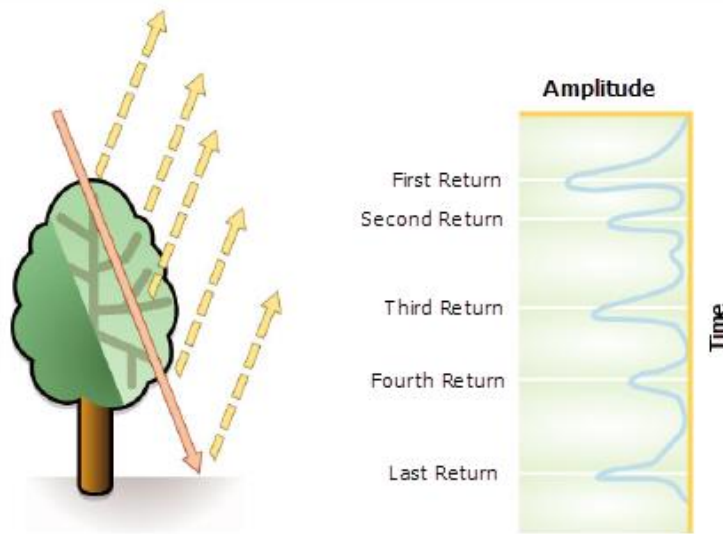
dönerek tarama yaparak ölçüm yapar. Alınan bu bilgiler dâhili bilgisayarda birleştirilerek haritalandırma gerçekleştirilir.

Lidar sistemlerinde genellikle iki tür algılama metodu kullanılır: doğrudan enerji algılama ve faz uyumlu algılama metotlarıdır. Faz uyumlu algılama metodu faz ve doppler etkisinin ölçümünde oldukça kullanışlı olup daha düşük enerjili ışınları kullanan karmaşık bir yapıya sahiptir.

Lidar sistemlerinde lazer darbeleri zemin yüzeyindeki cisme göre 1 veya daha fazla yansıma yapabilmektedir. Bu yansıma sayısı 1 ile 5 arasında olabilmektedir. Yansıma sayısı lazer darbesinin çarptığı cisim hakkında ayrı bir bilgi vermektedir.

İlk geri dönen lazer darbesi kullanıcı açısından en önemli olan lazer darbesidir. Bu geri dönüş muhtemelen ağaç tepesi, bina çatısı gibi yüksekliği fazla olan yerlerin birinden olmaktadır ama ilk geri dönüş zemin yüzeyinden de olabilmektedir. Eğer ilk geri dönüş zemin yüzeyinden olursa LİDAR sistemi sadece bir geri dönüş algılayacağı manasına gelmektedir.

Çoklu geri dönüşlerde ise lidar sistemi, lazer darbesi çarpan cismin yüksekliği hakkında bilgi vermektedir ve bu işlem ara yansımalarla göre hesaplanmaktadır. Ara dönüşlerde, bitki örtüsü veya toprak arazisi son dönüş olarak kullanılmaktadır.



Şekil 2.3: Lidar lazer dönüş sayısı [4]

Son dönüş her zaman bir toprak yüzeyinden olmayacaktır. Örnek verilecek olursa zemine gitmekte olan bir sinyal gidiş esnasında bir dal parçasına çarpıp geri yansıma yapıp zemine ulaşamayabilir. Bu durumda son geri dönüş zeminden değil ağaç parçasından olacaktır. Lazer dönüş sayısının görselleştirilmiş haline Şekil 2.3'den ulaşılabilir.

Lidar'ın başlıca kullanım alanlarını şu şekilde sıralanabilir:

- Yüzey analizleri,
- Sayısal arazi modeli oluşturulması,
- Sayısal yükseklik modeli oluşturulması,
- Sınıflandırma çalışmaları,
- Otomatik ayrıştırma/obje yakalama,
- 3B obje sayısallaştırma/modelleme,
- Ölçüm (mesafe, alan, hacim hesaplamaları).

Oluşturulan lidar lazer nokta bulutu x, y, z konum bilgileri ile saklanmaktadır. Bunların dışında; intensity, return number, return of number, point classification, edge of flight line, RGB (kırmızı, yeşil ve mavi) değerler, GPS saati, tarama açısı ve tarama yönü bilgileri de saklanmaktadır.

Bu bilgilerin birçoğu sınıflandırma işlemi sırasında kullanılması gereken önemli özellikler olarak karşımıza çıkacaktır. Tablo 2.1'den, her bir lidar noktasıyla sağlanabilecek özellikler açıklanmaktadır.

Tablo 2.1: Lidar için depolanan özellikler ve açıklamaları [4]

LİDAR ÖZELLİK	AÇIKLAMA
Intensity	Lidar noktasını oluşturan lazer darbesinin geri dönüş gücünü yansıtmaktadır. Yansıma yapılan yüzeye göre değişmektedir.
Return Number	Yayılan bir lazer darbesi, yansıtıldığı özelliklere ve verileri toplamak için kullanılan lazer tarayıcısının özelliklerine bağlı olarak 5 taneye kadar geri dönüşe sahip olabilir. İlk geri dönüş, bir geri dönüş numarası, ikinci geri dönüş numarası iki olarak işaretlenecektir.
Number of Returns	İade sayısı, verilen bir sinyal için toplam iade sayısıdır. Örneğin, bir lazer verisinin toplam beş geri dönüş sayısı için number of return değeri iki olabilir.
Point Classification	Son işlemde geçirilen lazer darbesi yansıtan nesne türünü tanımlayan bir sınıflandırmaya sahip olabilir. Lidar noktaları, çıplak toprak, gölgelik ve su dahil olmak üzere birkaç kategoride sınıflandırılabilir. Farklı sınıflar, farklı sayısal kodlar kullanılarak ifade edilmektedir. Görselleştirme sırasında da farklı renkler ile ifade edilir.
Edge of Flight Line	Noktalar, 0 veya 1 değerine göre sembolize edilecektir. Kenarları ifade eden noktalara 1 değeri geri kalan noktalara 0 değeri verilmektedir.
RGB	Lidar verileri RGB (kırmızı, yeşil ve mavi) bantlarla ilişkilendirilebilir.
GPS Time	Lazer noktasının yayıldığı zamanı ifade etmektedir.
Scan Angle	Tarama açısı -90 ila +90 derece arasında bir değerdir. 0 derece uçağın tam altını ifade etmektedir. -90 derece, uçağın sol tarafına, +90 ise lazer darbesi uçağın sağ tarafına doğru yönlendirilmektedir. Çoğu lidar sistemi +/-30 derece arasında tarama yapmaktadır.
Scan Direction	1 değeri pozitif tarama yönünü ve 0 değeri ise negatif tarama yönünü ifade etmektedir. Pozitif değer, uçuş yönünün soldan sağa, negatif değer ise sağdan sola tarama yapıldığını ifade etmektedir.

2.1 Lidar Nasıl Çalışır?

Lidar sistemlerinin çalışma prensibi gönderilen lazer darbesinin geri dönüş hesabının yapılmasıyla gerçekleşmektedir. Örnek verecek olursak, bir futbol topunu 30 metre uzaktaki bir duvara yere paralel olarak atıldığını varsayalım ve topun geri dönme süresini de 6 saniye olarak belirleyelim. Daha sonrasında duvara rastgele yaklaşıldığını ve topun tekrar aynı kuvvetle fırlatıldığını varsayalım. Bu atışta top 4 saniyede geri gelmekteyse, bu değişimden çıkan sonuç şudur ki duvara 10 metre yaklaşmış ve top o mesafeden fırlatılmıştır.

Yukarıda bahsedilen durumun yerine sabit hızla hareket eden bir uçak, sabit ışık hızı ile yayılan lazer darbesi göndererek yansımalar almaktadır. Geri dönen lazer darbeleri de futbol topunun geri dönmesindeki işlemlerdeki gibi hesaplanarak işleme sokulmaktadır. Lazer darbelerinin geri dönüş sürelerine göre topoğrafya hakkında sonuçlar elde edilmektedir.

2.2 Nokta Bulutu

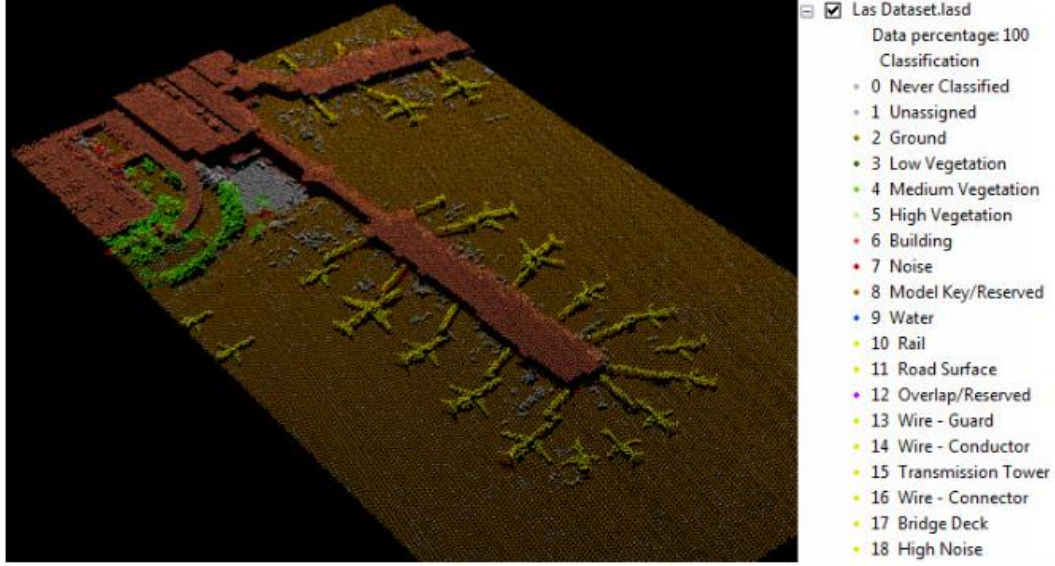
Lazer Tarama cihazlarından gönderilen lazer ışınının yüzeylere çarpıp yansması ile elde edilen x, y, z koordinat değerlerini içeren nokta kümesidir. Lidar veri setleri ağırlıklı olarak .las uzantılı 'binary' bir dosya formatında depolanır. Son işlemten geçmiş mekansal olarak düzenlenmiş lidar verileri, nokta bulutu verileri olarak bilinir. İlk nokta bulutları, GPS zaman damgaları gibi ek niteliklerle birlikte, x, y ve z koordinatlarını içeren büyük 3 boyutlu yükseklik nokta koleksiyonlarıdır.

Lazerin karşılaştığı spesifik yüzey özellikleri, ilk lidar nokta bulutu işlendikten sonra sınıflandırılır. Zemin, binalar, orman gölgesi, otoyol üst geçitleri ve araştırma sırasında lazer ışınının karşılaştığı herhangi bir şey için yükseklikler, nokta bulutu verilerini oluşturur.

2.3 LAS Dataset

LAS dosyası, havadan alınan LİDAR verilerini depolamak için endüstri standardı bir ikili formattır. LAS dosyalarını yerel formatta, hızlı ve kolay bir şekilde incelemenize ve LAS dosyalarında bulunan lidar verilerinin ayrıntılı istatistiklerini ve alan kapsamını sağlamanıza olanak tanır. LAS veri kümesi, yüzey kısıtlamaları içeren özellik sınıflarına referansı da saklayabilir.

LAS dosyaları genel olarak görsel bir sonuç vermektedir. Bu tezde üzerinde çalışılan dosyalar sayısal .txt formatları olacaktır ve bu sayısal veriler görselleştirilerek, sınıflandırma işlemleri yapılmaktadır. LAS dataset formatında görselleştirmeye Şekil 2.4'den ulaşabilirsiniz.



Şekil 2.4: LAS dataset [5]

2.4 Lidar Işıklılık (Intensity) Değeri

Yoğunluk, her nokta için toplanan, lazer darbesinin dönüş gücünü gösteren bir ölçüdür. Kısmen, lazer darbesinin çarptığı nesnenin yansıtıcılığına dayanır.

Yansıtma özelliği, en yaygın olarak yakın kızılötesinde kullanılan dalga boyunun bir işlevidir. Geri dönüşlerin gücü, geri dönüşü yansıtan yüzey nesnesinin bileşimine göre değişir. Yoğunluk, özellik saptama ve çıkartmada, lidar nokta sınıflandırmasında ve mevcut olmadığında havadaki görüntülerin yerine geçmekte yardımcı olarak kullanılır. Yoğunluk görecelidir, tam olarak ölçülebilir değildir, bu nedenle aynı ölçüm yüksekliğinde benzer ışıklılık sonuçları elde edilemeyebilir. Bu tez kapsamında kullanılacak olan dosyalarda 4. sütun ışıklılık değerini ifade etmektedir ve bu değer 0-255 arasında değişmektedir.

Lidar verileriniz yoğunluk değerleri içeriyorsa, bu verilerden siyah beyaz görsel elde etmek mümkündür. ArcGIS, lidar verilerinden yoğunluklu görüntü oluşturma yeteneği sağlar.



Şekil 2.5: Işıklılık değerin görselleştirilmesi [6]

2.5 Lidar Nokta Bulutu Sınıflandırması

Her lidar noktası, lazer darbesini yansıtan nesnenin türünü tanımlayan kendisine atanan bir sınıflandırmaya sahip olabilir. Lidar noktaları, toprak, bina, su dahil olmak üzere birkaç kategoride sınıflandırılabilir. Farklı sınıflar, LAS dosyalarındaki sayısal tamsayı kodları kullanılarak tanımlanır.

Sınıflandırma kodları Amerikan Fotogrametri ve Uzaktan Algılama Derneği (ASPRS) tarafından tanımlanmıştır. Sınıflandırma kodlarına Tablo 2.2'den ulaşılabilir.

Tablo 2.2: Lidar verisi sınıflandırma sayısal kodları [6]

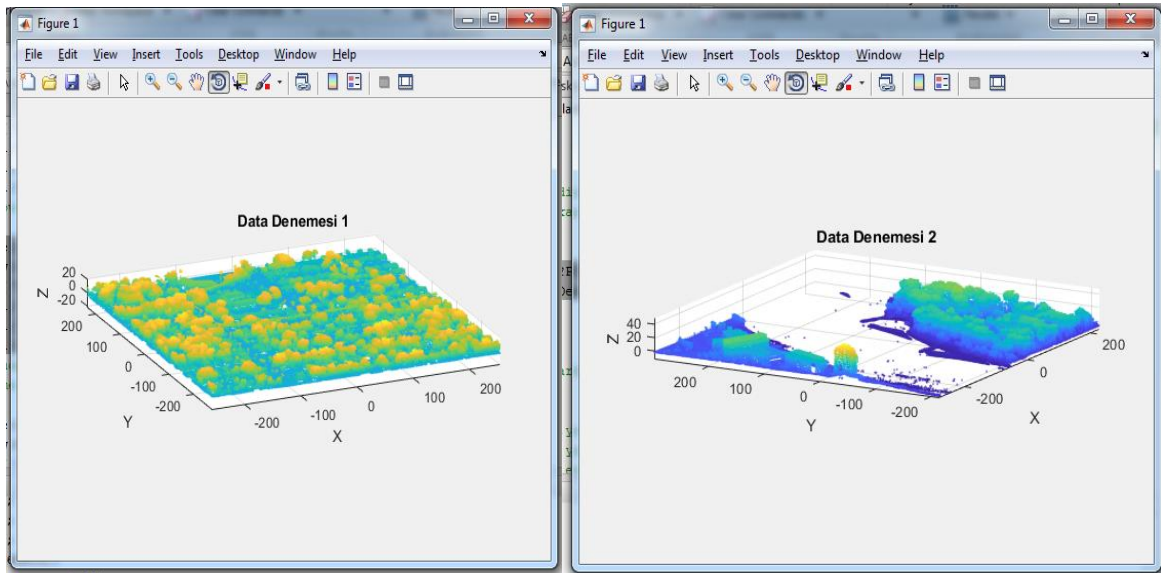
SINIFLANDIRMA KODU	SINIFLANDIRMA SONUCU
0	Sınıflandırılmamış
1	Atanmayanlar
2	Zemin
3	Düşük Bitki Örtüsü
4	Orta Bitki Örtüsü
5	Yüksek Bitki Örtüsü
6	Binan
7	Düşük Nokta
8	Ayrılmış
9	Su
10	Demiryolu
11	Yol Düzeyi
12	Ayrılmış
13	Tel Koruma
14	Tel İletken
15	İletim Kulesi
16	Tel Yapı Konnektörü
17	Köprü Güverte
18	Yüksek Gürültü
19-63	Ayrılmış
64-255	Kullanıcı Tanımlı

3. LİDAR VERİLERİNİN GÖRSELLEŞTİRİLMESİ

IEEE Data Fusion yarışmasında kullanılan veri setleri ilk olarak görselleştirme çalışması yapılmıştır. Bu çalışmada X, Y ve Z koordinatları kullanılarak Matlab yazılımı üzerinden çıktı alınmıştır. Bu çıktılarda dosyalarda var olan nokta değerleri görsel olarak anlamlandırılmıştır. Kullanılan veri setleri ortalama olarak 800.000 satır ve sınıflar da dahil olmak üzere 6 adet sütun içermektedir.

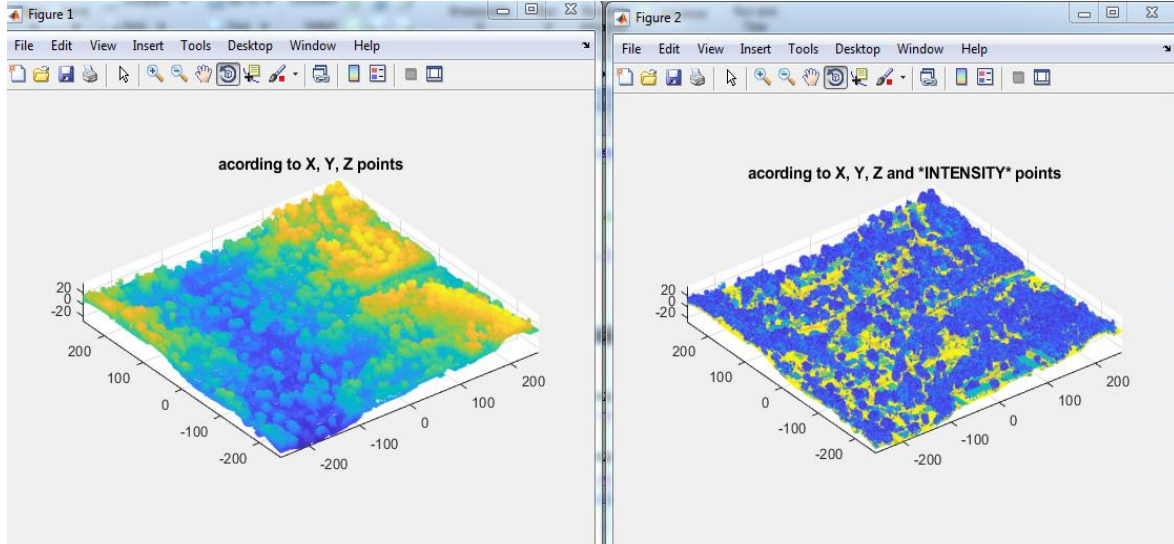
İndirilen veri setlerde ayrı ayrı birçok bölgenin veri koordinatları, ışıklılık ve dönüş sayısı değerleri bulunmaktadır. Toplamda 120 tane 5 sütun özelliği ve 110 tanede sınıf özelliği içeren veri seti bulunmaktadır. Bu tezde gözetimli öğrenme kullanılarak sınıflandırma işlemi yapıldığı için sınıflar dosyası var olan, etiketlenmiş hazır dosyalar üzerinden çalışmalar yapılmıştır.

Matlab’da yapılan ilk çalışma olan veri görselleştirilmesi için 2 ayrı dosya seçilmiştir ve bu dosyaların görselleştirilmiş ekran çıktılarına Şekil 3.1’den ulaşılabilir.



Şekil 3.1: Lidar nokta bulutu veri setlerinin X, Y, Z koordinatlarına göre görselleştirilmesi

Yapılan çalışmalarda X, Y, Z koordinatlarının görselleştirilmesinin haricinde ışıklılık değerinin de eklenerek 4 özniteliğin bir arada gösterilmesi Matlab üzerinden yapılmıştır. Şekil 3.1’de X, Y, Z koordinatlarına göre görselleştirme yapılırken, Şekil 3.2’de X, Y, Z koordinatlarına ışıklılık değeri de eklenerek görselleştirme işlemi yapılmıştır.



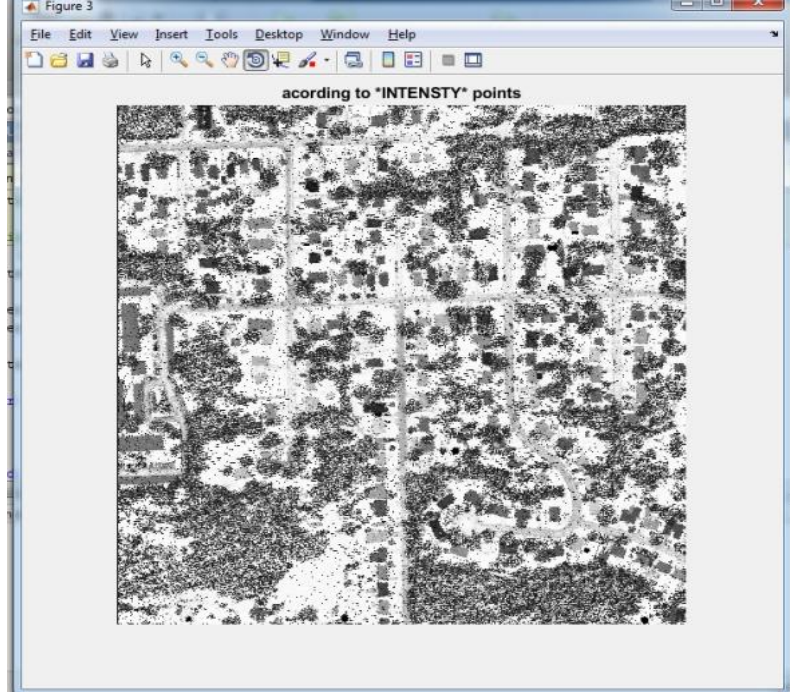
Şekil 3.2: Lidar verilerinin X, Y, Z koordinatları ve ışıklılık değerine göre görselleştirilmesi

3.1 Işıklılık (Intensity) Değerinin Görselleştirilmesi

İndirilen veri setleri de X, Y, Z koordinatları sınıflandırma için çok büyük önem arz etmesine rağmen ışıklılık değerleri de içerisinde yansıyan yüzey hakkında birçok bilgi içermektedir. Bundan dolayı ışıklılık değeri de kullanılan veri setlerinin sınıflandırılmasının önem arz eden özniteliklerden bir tanesidir.

Işıklılık değerleri 0-255 arasında olabilmektedir ve bu değer lazer ışınının çarptığı yüzey hakkında bilgiler barındırmaktadır. Şuana kadar yapılan görselleştirmeler de genelde X, Y, Z koordinatları üzerinden görselleştirme sonuçları renkli olarak elde edilmiştir. Bu sefer de Matlab yazılımı üzerinden sadece ışıklılık değeri siyah beyaz olarak görselleştirilmiştir. Beyaz olarak belirlenen yerler zeminden çok fazla yansıma yapılan yerleri ifade etmektedir. Işıklılık özniteliğinin görselleştirilmesine Şekil 3.3’den ulaşılabilir.

Bundan sonraki aşamalarda Matlab yazılımı ile farklı sınıflandırma yöntemleri incelenecektir. Kullanılan sınıflandırma türüne ve kullanılan öznelik sayısına göre sınıflandırmada elde edilen eğitim ve test başarımlar oranları incelenecektir.



Şekil 3.3: Işıklılık değerin görselleştirilmesi

4. MAKİNE ÖĞRENMESİNE GİRİŞ

Makine öğrenmesi (machine learning), matematiksel ve istatistiksel yöntemler kullanarak mevcut verilerden çıkarımlar yapan, bu çıkarımlarla bilinmeyene dair tahminlerde bulunan yöntemlerdir. [7]

Makine öğrenmesi esas olarak 1959 yılında bilgisayar biliminin yapay zeka da sayısal öğrenme ve model tanıma çalışmalarından geliştirilmiş bir alt dalıdır. Makine öğrenmesi yapısal işlev olarak öğrenebilen ve veriler üzerinden tahmin yapabilen algoritmaların çalışma ve inşalarını araştıran bir sistemdir. Bu tür algoritmalar statik program talimatlarını harfiyen takip etmek yerine örnek girişlerden veri tabanlı tahminleri ve kararları gerçekleştirebilmek amacıyla bir model inşa ederek çalışmaktadırlar.

Ayrıca makine öğrenimi, bilgisayarların algılayıcı verisi ya da veritabanları gibi veri türlerine dayalı öğrenimini olanaklı kılan algoritmaların tasarım ve geliştirme süreçlerini konu edinen bir bilim dalıdır. Makine öğrenimi araştırmalarının odaklandığı konu bilgisayarlara karmaşık örüntüleri algılama ve veriye dayalı akılcı kararlar verebilme becerisi kazandırmaktır.

Makine öğreniminin başlıca uygulamaları; makine algılaması, bilgisayarlı görme, doğal dil işleme, sözdizimsel örüntü tanıma, arama motorları, tıbbi tanı, beyin-makine ara yüzleri, kredi kartı dolandırıcılığı denetimi, borsa çözümlemesi, DNA dizilerinin sınıflandırılması, konuşma ve el yazısı tanıma, bilgisayarlı görmede nesne tanıma, oyun oynama, yazılım mühendisliği, uyarlamalı web siteleri, robot gezisi, yüz tanıma, belge sınıflandırma, otonom araçlar gibi sistemler makine öğrenmesine örnek olarak verilebilir.

Makine öğrenimi sistemlerinin bir bölümü insan sezgisine olan gereksinimi tümüyle ortadan kaldırmaya çalışırken bazıları insan ve makine arasında işbirliğine dayalı bir yaklaşım benimsemektedir. Sistemi tasarlayan kişinin verinin kodlanma biçimi üzerinde egemen oluşu, insan sezgisinin tümüyle ortadan kaldırılmasını imkansız hale getirmektedir. Makine öğrenimi deneysel yöntemin otomatikleştirilmesi olarak düşünülebilir.

Bazı istatistiksel makine öğrenimi araştırmacıları Bayes istatistiği çerçevesi kapsamında kullanılabilen yöntemler geliştirmektedirler. Makine öğrenmesinde ilk başta öğrenme süreciyle başlayarak ve test süreci ile devam etmektedir.

4.1 Terminoloji

- **Gözlem (Observations):** Öğrenmek ya da değerlendirmek için kullanılan her bir veri parçası. Örnek verilecek olursa, her bir e-posta bir gözlemdir. Değerlendirmeye alınacak her bir özellik parçası gözleme girmektedir. [8]
- **Öznitelik (Features):** Bir gözlemi temsil eden (genelde sayısal) verilerdir. Örnek verilecek olursa, e-posta'nın uzunluğu, tarihi, bazı kelimelerin varlığı. Gözlem verileri özellikler olarak isimlendirilmektedir. [8]
- **Etiketler (Labels):** Gözlemlere atanan kategoriler. Örnek verilecek olursa, spam-spam değil. Etiketler bir manada sınıflandırma verileri olarak da kullanılmaktadır. [8]
- **Eğitim Verisi (Training Data):** Algoritmanın öğrenmesi için sunulan gözlemler dizisi. Algoritma bu veriye bakarak çıkarımlarda bulunmaktadır ve bir model oluşturulmaktadır. Örnek verilecek olursa, çok sayıda spam-spam değil diye etiketlenmiş e-posta gözlemi. [8]
- **Test Verisi (Test Data):** Algoritmanın oluşturulan modelin ne kadar gerçeğe yakın olduğunu test etmek için kullanılan veri setidir. Eğitim esnasında saklanır, eğitim bittikten sonra etiketsiz olarak algoritmaya verilerek algoritmanın (vermediğimiz etiketler hakkında) tahminlerde bulunması beklenir. Örnek verilecek olursa, spam olup olmadığı bilinen (ama gizlenen), eğitim verisindekilerden farklı çok sayıda e-posta gözlemi. [8]

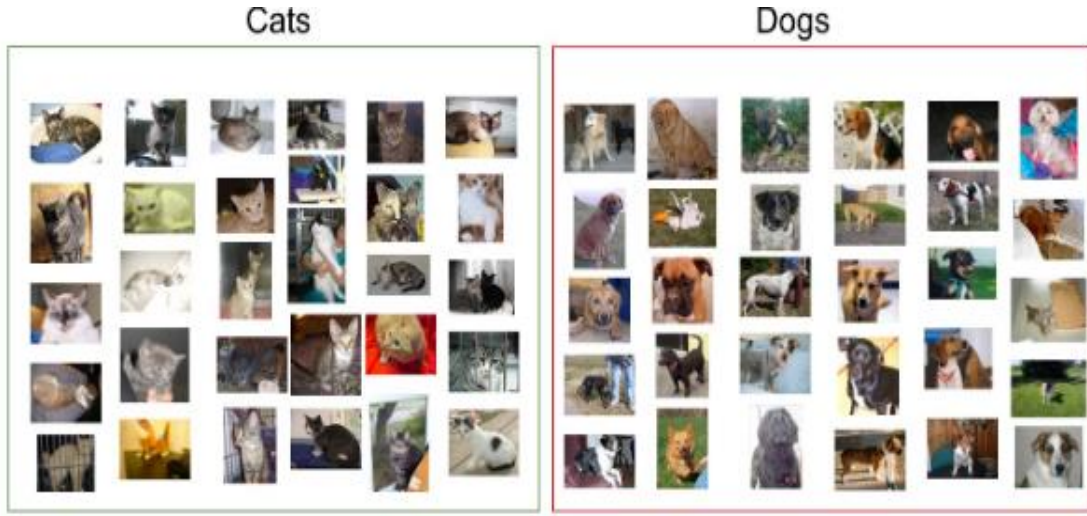
4.2 Öğrenme Yaklaşımları

4.2.1 Gözetimli Öğrenme (Supervised Learning)

Bu öğrenme modelinde etiketli veriler kullanılarak öğrenme işlemi gerçekleştirilmektedir. Öğrenme işlemi belli bir veri seti üzerinden etiketli veriler ile yapılmaktadır. Sistem etiketli verilerden öğrenme işlemi yaparak, yeni gelen verileri öğrendiklerine göre tahmin eder.

Bir sistem tasarlanacağını varsayılırsa ve eğer bu sistem gösterilen resimden kedi köpek olduğunun tahminini yapıyor ise böyle bir sistem için gözetimli öğrenme algoritmalarına ihtiyaç duyulmaktadır. Eğer resimde bir kedi olduğu algılanacaksa sistem öncelikli olarak

kedi resmi içeren veri seti ile eğitilmektedir. Bu veri seti eğitim verisi olarak isimlendirmektedir. Etiketlenmiş örnek eğitim verisine Şekil 4.1’den ulaşılabilir.



Şekil 4.1: Eğitim verileri [9]

Bu eğitim verisindeki fotoğrafların “kedi fotoğrafı” olduğunu eğitim verisinde belirtilmesi gerekmektedir, yani veri etiketlenmelidir. Bu ve benzer eğitim verileri ile sistem ve algoritma eğitilmektedir.

Eğitilen sistem, eğitim verileri haricinde etiketsiz veriler verilerek test işlemi gerçekleştirilir. Bu örnekte olduğu gibi bir eğitim veri seti ile eğittiğimiz makine öğrenmesi yöntemine gözetimli öğrenme yöntemi denilmektedir. Gözetimli öğrenme yöntemleri sınıflandırma ve regresyon olmak üzere iki şekilde incelenebilir.

4.2.1.1 Sınıflandırma (Classification)

Her bir gözleme bir kategori ve sınıf atması yapılır. Sınıflar ayrıktır (sayı değildir) ve birbirlerine yakın veya uzak olmaları gibi bir durum söz konusu değildir.

Sınıflandırma yapısal (structure) veya yapısal olmayan (unstructure) veriler üzerinde yapılabilir. Eğer sistem, hangi verinin hangi koşullarda hangi sınıfa ait olacağı bilgisi ile sınıflandırılarak eğitilirse, yeni veri setindeki veriyi de öğrendiklerine benzer biçimde sınıflandıracaktır.

Sınıflandırma Yöntemleri aşağıdaki gibi kategorize edilebilir:

- **İkili Sınıflandırma (Binary Classification):** İki olası sonucu olan sınıflandırmalardır. Örnek olarak cinsiyet sınıflandırması verilebilir.
- **Çoklu Sınıf Sınıflandırma (Multi Class Classification):** İkiden fazla sınıfı sınıflandırma işlemine verilen isimdir. Bir sınıfa ait birden fazla farklı veri vardır ve her bir farklı veri farklı sınıflara atanır. Örnek olarak bir hayvan sınıfında kedi ya da köpek olabilir ancak ikisi birlikte bir sınıfta olamaz ve kendi içinde de sınıflara bölünebilirler.
- **Çoklu Etiket Sınıflandırma (Multi Label Classification):** Bir veri birden fazla sınıfla ilişkilendirilebilir. Bir makalenin hem sağlık hem spor hem de insan ile ilgili olması bu sınıf için örnek verilebilir.

Sınıflandırma yönteminde en çok kullanılan algoritmalar ise şu şekilde sıralanabilir:

- **Naive Bayes:** Verileri olasılık ilkeleri ile hesaplayarak sınıflandıran bir sınıflandırma algoritmasıdır. Basit bir ifadeyle, bir Naive Bayes sınıflandırıcı, bir sınıftaki belirli bir özelliğin varlığının başka herhangi bir özelliğin varlığına bağlı olmadığını varsayar.

Eğitilmiş veriler üzerinde olasılık işlemleri yapılarak, sisteme sunulan yeni verilerin önceki olasılık değerine göre sınıflandırılması sağlanır. Örnek verilecek olursa, binlerce makalenin hangi alanda yazıldıklarına göre kategorize etmek isteniyorsa, bunun için belli makalelerde geçen belli kelimelerin olasılık değerlerinin, diğerlerine oranla fazla olması durumuna göre o makalenin hangi kategoriye ait olduğunu öğrenmek istenilir.

- **En Yakın Komsu (K-Nearest Neighbours):** Bu tip sınıflandırma, her bir noktanın en yakın komşularının basit çoğunluk oyu ile hesaplanması ile elde edilen sınıflandırmadır. Veri hangi veriye en çok yakındır mantığı ile çalışmaktadır. Bu algoritmanın uygulanması kolaydır, gürültülü eğitim verisine (noisy training data) dayanıklıdır.

- **Karar Ağacı (Decision Tree):** Veriler, sınıfları ile birlikte bu algoritmaya verildiğinde, algoritma verileri sınıflandırmak için kullanılabilecek bir dizi kural üretir. Karar düğümleri (decision node) ve yaprak düğümleri (leaf node) olan bir ağaç yapısına sahiptir. Hem sınıflandırma hem de regresyon yönteminde kullanılabilir.

- **Random Forest:** Sınıflandırma işlemi sırasında birden fazla decision-tree kullanılarak sınıflandırma değerinin yükseltilmesi hedefleyen, sınıflama veya regresyon yönteminde kullanılabilen algoritmadır.

- **Destek Vektör Makinesi (Support Vector Machine):** Veri setinde birbirine benzeyen gruplar arasına birbirinden en uzak olan noktalardan sınırlar çizmeye yarayan bir sınıflandırma algoritmasıdır.

4.2.1.2 Regresyon (Regression)

Her gözlem için öğrendiklerine bakarak reel bir değer tahmini yapmaktadır. Regresyon problemleri, üretilen çıktının sürekli sayılardan oluştuğu durumlar için kullanılmaktadır. "2011 model 40.000 km'de Mia Toronto arabanın fiyatı 45.670 TL olmalıdır" şeklinde bir çıkarım regresyona örnek verilebilir.

Regresyon algoritmaları gözetimli öğrenme kategorisinde olmasının yanı sıra durum yakalama gibi durumlarda hem gözetimli (supervised) hem gözetimsiz (unsupervised) öğrenme yöntemleriyle de kullanılmaktadır.

En çok kullanılan regresyon algoritmaları şunlardır:

- **Linear Regression:** Sayısal girdi ve çıktılar arasındaki doğrusal ilişkiyi tespit etmeyi sağlar. Düzlemde yayılmış verinin modelini en iyi biçimde doğrusal olarak çıkartmaya çalışan yöntemdir.

- **Logistic Regression:** Bir sonucu belirleyen bir veya daha fazla bağımsız değişken bulunan veri kümesini analiz etmek için düzlemde en iyi eğriyi yakalamaya çalışan istatistiksel bir yöntemdir. Sonuç, ikiye bölünmüş bir değişkenle ölçülür (sadece iki olası sonuç vardır).

- **Multiple Linear Regression:** Birden fazla tahminleyici (predictor) değişken kullanarak tahminlemeye çalışılan doğrusal regresyondur.

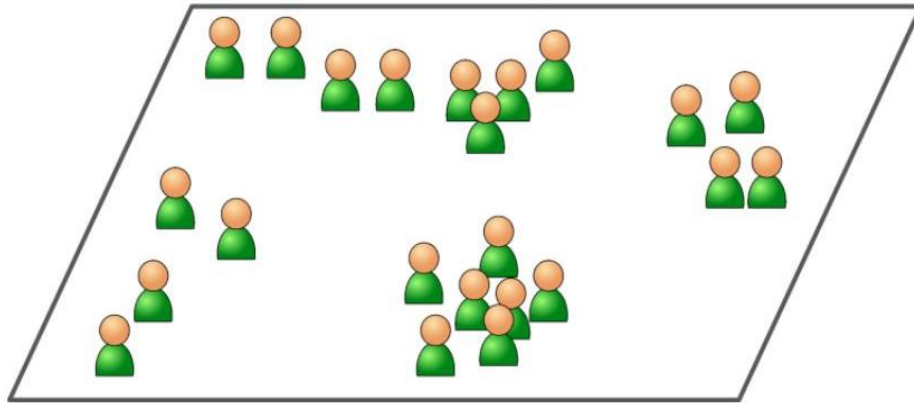
- **Polynomial Regression:** Veriler arası ilişki her zaman doğrusal olmayabilir. Optimum ilişkiyi bulmak için bir eğri gerekebilir. Tıpkı polinom fonksiyonlarında olduğu gibi bu yöntemde de bir terimin karesi, küpü vs. alınarak doğrusal olmayan bir regresyon modeli oluşturulmak istenebilir. Bu gibi durumlarda kullanılabilen bir algoritmadır.

- **Support Vector Regression:** Algoritmayı karakterize eden tüm ana özellikleri (maksimal marjı) koruyan bir regresyon yöntemi olarak da kullanılabilir. SVM ile aynı ilkeleri kullanır. Ana fikir, hatanın en üst düzeye çıkarıldığı hiper düzlemi bireyselleştirerek hatayı en aza indirmek, hatanın bir kısmının tolere edildiğini göz önünde bulundurmaktadır.

- **Decision Tree:** Sınıflandırma yönteminde de kullanılan decision tree, regresyon yönteminde de aynı şekilde kullanılabilir. Bu algoritma, kök düğümünden başlayarak, yukarıdan aşağıya inşa edilen düğümler (node) ile verilerin, kendi içlerinde benzer değerlere (homojen) sahip olanlarının alt kümelerine ayrılmasını sağlayan algoritmadır.

4.2.2 Gözetimsiz Öğrenme (Unsupervised Learning)

Gözetimsiz öğrenme, gözetimli öğrenme yönteminin aksine herhangi bir kategorize edilmiş, etiketlenmiş eğitim verisi kullanılarak eğitilmemektedir. Gözetimsiz öğrenme yöntemi, önceden eğitilmemiş veriler üzerinde çalışarak veriler arasında bağıntılar bulup birbirine yakın anlamda, içerikte, değerde olan verilerin kendi içinde kümelenmesi mantığıyla çalışmaktadır.



Şekil 4.2: Gözetimsiz öğrenme yöntemi [10]

Gözetimsiz öğrenme bir girdi kümesi modeller ve etiketsiz gözlemlerden öğrenme sürecidir. Algoritmanın kendi kendine keşifler yapmasını ve gözükmeyen örüntüleri keşfedilmesi bu öğrenme metodunda beklenmektedir.

Girdi verisinin hangi sınıfa ait olduđu önceden bilinmemektedir. Bu sınıflandırma işlemleri veriye bakılarak algoritmalar tarafından öğrenilmektedir. Yeni gelen veriler de algoritmanın oluşturduđu gruplara uygun olarak en yakın gruba atanmaktadır.

Bir marketteki müşterilerin ürün satın alırken davranışlarını incelemek ve ona uygun şekilde ürünlerin stoklarını güncellemek veya raf düzeni yapmak istenildiği varsayalım. Her gün yüzlerce müşterinin geldiğini göz önünde bulundurursak, bu durumda sistemi “bu satılırsa şu da satılır” şeklinde bir eğitim verisi ile eğitilmesi imkansız olacaktır.

Bu yüzden gözetimli öğrenme algoritmaları kullanmak yerine gözetimsiz öğrenme algoritmaları kullanarak, birbiri ile yakın alışveriş alışkanlıkları olan müşteri gruplandırılabilir. Çıkan gruplara göre stoklar güncelleyebilir veya raf düzenlemesi buna uygun yapılabilir. Bu şekilde gözetimsiz öğrenme tekniği kullanılmış olur. Kısacası eğitim verisi hazırlanamayacak karmaşıklıkta veriler için gözetimsiz öğrenme yöntemi kullanılmalıdır.

Gözetimsiz öğrenme kümeleme (clustering), birliktelik kuralı (association rule mining) ve boyut azaltımı (dimensionality reduction) olarak 3 başlık altında incelenebilir.

4.2.2.1 Kümeleme (Clustering)

Kümeleme yöntemi, veri setindeki her bir verinin birbirlerine benzerlik durumlarına göre gruplara ayrılması işlemine denir. Gözlemler homojen bölgelere ayırır.

Her bir grup birer küme anlamına gelir. Her kümede birbirine en yakın veriler yer alır. Birbiriyle benzerlik göstermeyen veriler de mümkün olduğunca farklı kümelerde yer alırlar. Kısacası kümeler arası benzerliğin az olması kümeleme için iyi bir durumdur ama çok fazla küme ile çalışmak da birbirine çok benzeyen kümelerin oluşmasına sebep olabilir. Bu yüzden optimal bir küme sayısı belirtmek gerekir. Ancak küme sayısını kolayca belirlemek her veri seti için mümkün olmamaktadır. Bu yüzden K-Means Algoritması kullanılır. Bu algoritma, kullanacağınız veri setinde oluşturulacak optimal küme sayısını vererek sonuçların sağlıklı olmasını sağlamaktadır. [10]

4.2.2.2 Boyut Azaltımı (Dimensionality Reduction)

Boyut azaltma, amaç doğrultusunda en iyi sonucu verecek olan özniteliklerin (feature) belirlenmesini sağlamaktadır. Kullanışsız, gereksiz olan öznitelikleri çıkartmak, veri boyutunu azaltıp işlem yükünün azaltılması olarak nitelendirilebilir. Daha az öznitelik ile çalışmak işlem hızını ve sağlıklı sonuç alma oranını genel olarak arttırmaktadır.

Örnek olarak verilecek olursa bir kişinin kan değerlerine bakılarak bir hastalık tespiti yapmak isteniyorsa, buradaki kan değerlerinin her birine ihtiyaç olacaktır ama kişinin saç, göz ve ten rengi gibi özniteliklere ihtiyaç yoktur.

Gereksiz öz nitelikleri bu veri setinden atıp daha az veri ile çalışmalar yapılabilir. Bu işlem “feature selection” olarak isimlendirilmektedir. Ayrıca veri boyutu çok büyük ve hangi verinin nitelikli olduğu konusunda tam bir sonuç elde edilemiyorsa yoksa PCA (Principal Component Analysis) algoritması ile bu işlemleri yapmak için kullanılabilir.

4.2.2.3 Birliktelik Kuralı (Association Rule Mining)

Bu kural kümeleme yönteminden farklı olarak değişkenler arası ilginç ilişkileri keşfetmek için kurallar arayan bir yöntemdir.

Veri seti içindeki geçmiş tarihli hareketlerin örüntülerini analiz eden ve birlikte gerçekleşme durumlarını çözümleyen veri madenciliği yöntemidir. Bu örüntülerden hareketle gelecekteki veriler için tahminler yapabilmektedir. Bu durum sebebiyle regresyonu anımsatmaktadır.

Birliktelik kuralı (association rule) algoritmaları ile müşteri-sepet analizi yapılabilir. Hangi müşteri hangi ürünle birlikte neleri satın almış bunları incelenip, bu sayede geliri arttırmak ve verimi yükseltmek adına her müşteri için özel ürün tavsiyelerinde bulunulabilir. Promosyonlar yapılabilir, stok bu bilgiler ile güncel tutabilir. Raf düzenlemesinde de bu analizlerden faydalanabilir, ilişkili ürünleri birbirine yakın dizilebilir.

Apriori, eclat, FP-growth algoritmaları, birliktelik kuralı için en çok tercih edilen algoritma türleridir. Bu algoritmaların hepsi kendi içerisinde farklı matematiksel işlemler içermektedir.

Sonu olarak, kmeleme veri noktaları ile ilgilenmekteyken, birliktelik kuralı bu veri noktalarının nitelikleri arasındaki iliřkileri bulmakla ilgilenir.

4.2.3 Takviyeli ğrenme (Reinforcement Learning)

Bu ğrenme biimi temelinde canlıların davranıř psikolojisi yer almaktadır. Bu yntem ğrenme iřlemine evreden aldıėı geri bildirimler (feed-back) ile gerekleřtirmektedir. Bu ğrenmede, gemiřten gelen birikimlerle denemeler yapılarak hedefe ulařılmazsa ceza (penalty) sinyali, hedefe ulařıldıėında ise dl (reward) sinyali alınmaktadır.

Eėer sistem ceza sinyali alırsa bu iřlemi tekrar gerekleřtirmemektedir. dl sinyali alındıėında ise ğrenmeye devam etmektedir. Bundan dolayı sistem srekli ğrenmeye devam ederek, sistem algoritması mkemmeli yakalamaya alıřmaktadır.

4.3 Makine ğrenmesi Adımları

Makine ğrenmesi genel olarak veri toplama, verileri hazırlama, model seimi, modelin eėitimi, deėerlendirme, parametre ayarı, tahmin-test gibi bir ok ařamadan oluřan kompleks bir yapıya sahiptir. Genel hatları ile adımlar ařaėıda gibi aıklanabilir. [11]

4.3.1 ğrenme Sreci

ğrenme srecinde bir model oluřturmadan nce birok zelliėe gre ayrı sınıflar oluřturulmaktadır. Kullanılacak veri setleri internet zerinden .raw dosyası veya .txt dosyası olarak ulařım saėlanabilir.

4.3.2 zellik ıkarımı

Gzlemlerimizi en iyi temsil edecek zellikleri kullanmak iin ham veriden znitelik ıkartımı (feature extraction) ve ıkarılan zellikler zerinde zellik seimi (feature selection) yapılmaktadır.

Bu srete alan bilgisi (domain knowledge) uygulamak gerekmektedir. Yani bu iři tecrbesiz bir insana anlatacak olsaydıė hangi zelliklerin anlatılması daha etkili olurdu sorusu sorularak

özellik kullanılacak özellik seçimi yapılmalıdır. Bu özellikleri tanımlayıp hesaplamak gerekmektedir. Kısacası özellik çıkartımı veri setinin gereksiz kısımlarını atmaktır.

Bu yapılan tezde kullanılan veri setinde 6 öznitelik bulunmaktadır. Lidar veri seti sınıflandırma yapılırken verilerin sınıflandırmasında en önemli özellikler Z koordinat ve ışıklılık bilgisi olarak ele alınmıştır.

4.3.3 Model Oluşturma

Özellikler çıktıktan sonra bir öğrenme algoritması bu matris üzerinde çalışmaktadır. Elde edilen sonuca göre de bir takım kurallar çıkarılır ve bu kurallar dahilinde de model oluşturulur.

4.3.4 Değerlendirme-Test

Elde edilen modelin farklı test verileri ile çalışmasının gözlemlenmesidir. Test sonucunda oluşturulan modelin başarı oranı gözlemlenmektedir. Bu orana göre de oluşturulan sistemin ne kadar verimli çalıştığının yorumu yapılmaktadır. Değerlendirme sürecinde dikkat edilmesi ve göz önünde bulundurulması gereken bazı parametreler bulunmaktadır.

4.3.4.1 Değerlendirme-Test Sürecinde Kullanılan Parametreler

- **Doğruluk (Accuracy):** Oluşturulan modelde elde edilen veriler ve sonuçlarında yüzde kaç doğruluk elde edildiğini gösteren parametredir. Kısacası doğru tahmin edilen etiketlerin, tüm test etiketlerine oranı olarak tanımlanabilir.
- **Hassaslık (Sensitivity):** Modelin tam olarak hiçbir fikri olamadığı zaman elde edilen sonuca verilen isimdir. Kısacası hiçbir fikir olmadığı zaman öne sürülecek en iyi cevaptır.
- **Belirginlik (Specificity):** Oluşturulan modelde belli bir sınıftan yüzde kaç doğru olarak tahmin edildiği ile alakalı bir durumdur.

Modelin karmaşıklığına ve kullanılan veri sayısına göre elde edilecek sonuçlar değişecektir. Eğer model çok basitse elde edilecek sonuçlar çok hızlı çalışacaktır. Eğer iyi genelleme

yapabiliyorsa farklı veri kümelerinde iyi tahminler yapılabilir. Ancak az miktarda veriye bakılarak genelleme yapıldıysa, bu genelleme farklı veri kümelerinde yanlış tahminlere yönlendirme yapabilmektedir.

Modelin karmaşıklığını artırılırsa bu sefer sistem eğitim kümesini ezberlemeye çalışmaktadır. Bu modeli eğitim kümesinde test edecek olursak %100'e yaklaşan başarı gösterecektir. Ancak test verisinde farklı bir sınıf gelmesi durumunda sistem sıkıntı yaşayacaktır ve hatalı çıktılar elde edilecektir. Bu durumda modelin karmaşıklığı iyi ayarlanmalıdır. Model ne eğitim verisini ezberlemeli, ne de çok basit genellemeler yapmalıdır. Mümkün olduğunca (fazla değil) basit ve seçiciliği yüksek veriler üzerinde çalışmalar yapılmalıdır.

4.3.5 İyileştirme

Eğer oluşturulan modelin tam olarak başarılı olmadığı düşünülüyorsa, nerelerde hata yapılabileceği tespit edilerek daha seçici bilgilerle model tekrar eğitilebilir. Elde kalan yeni bilgilere göre de yeni bir model oluşturularak, tekrar değerlendirme sürecine gidilebilir.

İyileştirme sürecinin sonu yoktur. Tatmin edici noktaya varana kadar iyileştirme işlemine devam edilebilir. Ancak %100'lük başarı oranı yakalamak imkansız gibi bir şeydir.

4.3.6 Tahmin

Eğer model test kümesinde başarılı şekilde tahmin yürütebiliyorsa, başka veriler üzerinde çalıştırıp tahminler yapılarak, modelden faydalanmaya başlanabilir.

5. KULLANILAN SINIFLANDIRMA YÖNTEMLERİ

Bu bölümde bitirme tezinde sıkça kullanılan SVM ve KNN algoritmaları incelenecektir. 4. bölümde SVM ve KNN ile alakalı bilgiler sınıflandırma çatısı altında yüzeysel olarak verilmişti ama bu bölümde biraz daha detaylı ve anlaşılır bir şekilde anlatılmaktadır.

5.1 Destek Vektör Makineleri (SVM)

Sınıflandırma için kullanacak ilk yöntem SVM olarak belirlenmiştir. SVM, 1963 yılında Vladimir Vapnik ve Alexey Chervonenkis tarafından temelleri atılan istatistiksel öğrenme teorisine dayanan gözetimli öğrenme algoritmasıdır. [12]

SVM iki veya daha fazla çeşit verinin belli özelliklere göre sınıflandırılmasını sağlayan, makine öğrenme yöntemlerinden birisidir. Makine öğrenmesi, tahmin etmeyi ve sınıflandırmayı içermektedir. Bu amaçla kullanılacak olan veri setine göre farklı makine öğrenme algoritmaları kullanılmaktadır. İlerleyen bölümlerde SVM ve KNN dışında SaCR yöntemleri kullanılarak da testler ve sınıflandırma başarımları incelenecektir.

İncelenen sınıflandırma yöntemlerinde öznitelik sayısı değiştirilerek elde edilen test başarımlarının artırılması hedeflenmiştir. Elde edilen sınıflandırma sonuçlarının incelenmesine geçmeden önce kısaca SVM ve KNN incelemesi yapılacaktır. Bu tezde ise sayısal Lidar veri setlerinin sınıflandırılması için kullanılacaktır.

SVM sınıflandırma ve regresyon sorunları için doğrusal bir modeldir. Doğrusal veya doğrusal olmayan sorunlara çözümler üretebilmektedir. SVM'in ana mantığı veri sınıflarını ayıran bir çizgi veya hiper düzlem oluşturmaktır. Ayrıca SVM ile önceden etiketlenmiş verilere eğitim yapılarak, test sırasında yeni gelecek olan verilerin sınıflandırmasını yapılmaktadır.

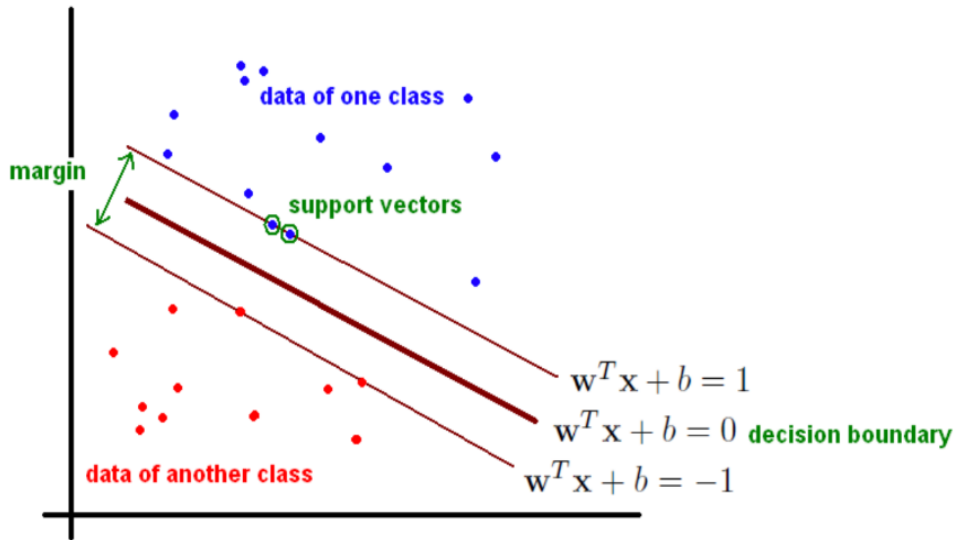
SVM, sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayrılmaktadır. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olarak belirlenir. İşte SVM bu sınırın nasıl çizileceğini belirleyerek bu işlemin yapılması için iki gruba da yakın ve birbirine paralel iki sınır çizgisi çizer ve bu sınır çizgilerini birbirine yaklaştırılarak ortak sınır çizgisi üretilir.

Basit anlamda sisteme giren her girdinin (input) bir özellik çıkarımı (feature extraction) yapılmış ve sonuçta bu iki boyutlu düzlemde her girdiyi gösteren farklı bir nokta elde edilir. Elde edilen noktalar sınıflandırılarak veri seti sınıflandırılmış olmaktadır.

Veri setini ikiye ayıran doğru “karar doğrusu” olarak isimlendirilmektedir. Sonsuz tane karar doğrusu çizebilme imkanı mevcut olsa da önemli olan en uygun karar doğrusunu belirlemektir.

Destek vektörlerinin üzerinde bulunduğu düzlemlere “sınır düzlemleri” denir. Sınır düzlemlerinin tam ortasından geçen ve her iki düzleme de eşit uzaklıkta bulunan düzlem ise “hiper düzlem” olarak ifade edilir.

Karar doğrusunun yeni katılacak olan veriye karşı dayanıklı olabilmesi için sınır çizgisinin, iki sınıfın sınır çizgilerine en yakın uzaklıkta olması gerekmektedir. Bu sınır çizgisine en yakın noktalar, “destek noktaları” olarak adlandırılır ve sınıflar genelde -1 ve +1 olarak isimlendirilmektedir. [2] SVM genel yapısına Şekil 5.1’den ulaşılabilir.



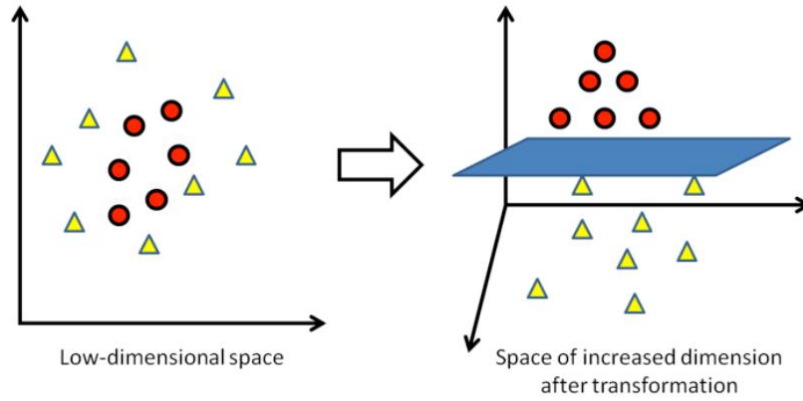
Şekil 5.1: SVM genel yapısı [13]

SVM’de birçok çeşit kernal kullanılmasına rağmen yapılacak uygulamalar da kernal boyutları “Auto” olarak belirlenerek daha başarılı sonuçlar elde edilmesi sağlanmıştır. Veri setine göre kullanılacak SVM başarımları oranı değişiklik göstermektedir. Sık kullanılan SVM türlerine Şekil 5.2’den ulaşılabilir.

Type of SVM	Mercer Kernel	Description
Gaussian or Radial Basis Function (RBF)	$K(x_1, x_2) = \exp\left(-\frac{\ x_1 - x_2\ ^2}{2\sigma^2}\right)$	One class learning. σ is the width of the kernel
Linear	$K(x_1, x_2) = x_1^T x_2$	Two class learning.
Polynomial	$K(x_1, x_2) = (x_1^T x_2 + 1)^\rho$	ρ is the order of the polynomial
Sigmoid	$K(x_1, x_2) = \tanh(\beta_0 x_1^T x_2 + \beta_1)$	It is a mercer kernel for certain β_0 and β_1 values only

Şekil 5.2: SVM sınıflandırma çeşitleri [14]

SVM ile doğrusal veriler sınıflandırıldığı gibi doğrusal olmayan verilerin sınıflandırılması da yapılmaktadır. SVM ile sınıfları birbirinden ayıran en iyi çizgi bulunmaya çalışılmaktadır. Çizilecek doğru sınıfların hepsine en uzak olacak şekilde ayarlanması gerekmektedir. Bazı durumlarda veri seti doğrusal olmayabilir. Bu durumda doğrusal olmayan verilerin sınıflandırılması için çekirdek hilesi'ne (kernel trick) başvurulmaktadır. Polynomial kernel ve gaussian RBF (Radial Basic Function) kernel en çok kullanılan çekirdek hilesi yöntemlerindendir. Bu durumda doğrusal olmayan bir veri, çekirdek hilesi ile 2 boyuttan 3 boyuta geçirilerek doğrusal hale dönüştürülmektedir. Çekirdek hilesi ile alakalı görsele Şekil 5.3'den ulaşılabilir.



Şekil 5.3: Çekirdek hilesi ile 2 boyutlu grafiği 3 boyutlu hale getirmek [15]

Ayrıca yapılacak olan uygulamalarda SVM ile çoklu sınıflandırmalar yapılacaktır. Çoklu sınıflandırma işlemlerin de sınıflandırılacak veriler ikili sınıflar haline çevrilebilir veya sınıflandırma işlemi bir grup üzerinden tüm sınıfların ayrılması şeklinde yapılabilir. Bu tez çalışmasında çoklu SVM çoklu sınıflandırma için Matlab Classification Learner'dan yararlanılarak linear ve gaussian SVM türlerinde incelemeler yapılmıştır. Yapılan işlemlere ve sonuçlarına ilerleyen bölümlerden ulaşılabilir.

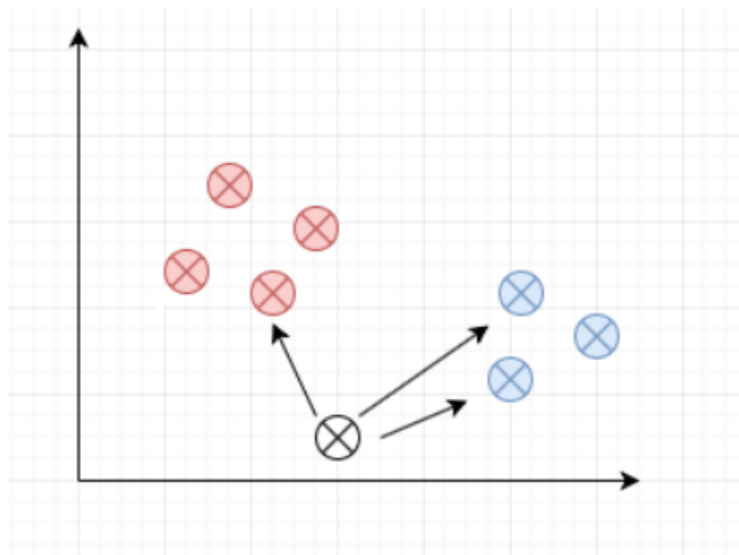
5.2 K-En Yakın Komşuluk (KNN, K-Nearest Neighbor)

KNN gözetimli öğrenme yöntemlerinde, SVM gibi sınıflandırma işlemi için kullanılan yöntemlerden birisidir. Sınıflandırmanın haricinde KNN algoritması regresyon işlemi için de kullanılabilir. KNN algoritmaları, 1967 yılında T. M. Cover ve P. E. Hart tarafından önerilmiştir. Algoritma, sınıfları belli olan bir örnek kümesindeki verilerden yararlanılarak kullanılmaktadır. [15]

KNN, parametrik olmayan (non-parametric) bir öğrenme algoritmasıdır. Eğitim verilerini öğrenmez, bunun yerine eğitim veri kümesini ezberler. Bir tahmin yapmak istediğimizde, tüm veri setinde en yakın komşuları arar. [4]

Burada kullanılan k değeri alınan noktaya en yakın k değere bakmak için kullanılmaktadır. K değeri sonucunda karşılaştırma yapılarak sınıflandırma türüne karar verilmektedir. Genellikle k değerleri eşitlik olmaması adına tek sayılar seçilmektedir.

En yakın noktaları bulmak için uzaklık hesabı formüllerinden yararlanılmaktadır. Uzaklık hesaplama işleminde genelde Öklid fonksiyonu kullanılır. Öklid fonksiyonuna alternatif olarak Manhattan, Minkowski ve Hamming fonksiyonları da kullanılabilir. KNN algoritmasının genel çalışma mantığına Şekil 5.4’de ki görselden ulaşılabilir.



Şekil 5.4: KNN örnek çizimi [15]

Ayrıca KNN eski, basit ve gürültülü eğitim verilerine karşı dirençli olması sebebiyle en popüler makine öğrenme algoritmalarından biridir. Fakat bunun yanında dezavantajı da mevcuttur. Örneğin, uzaklık hesabı yaparken bütün durumları sakladığından, büyük veriler için kullanıldığında çok sayıda bellek alanına gereksinim duymaktadır.[15]

KNN algoritmasının çalışma adımlarına Şekil 5.5'den ulaşılabilir.



Şekil 5.5: KNN çalışma adımları [16]

KNN algoritmasının adımları:

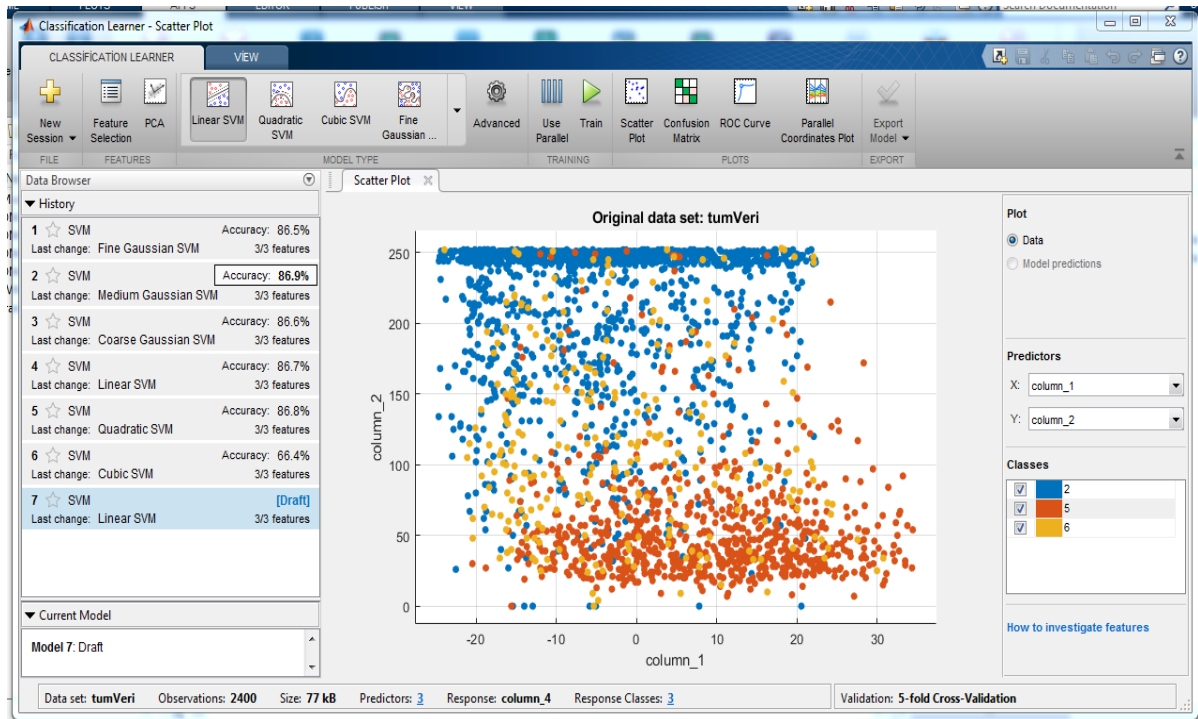
- İlk olarak k parametresi belirlenir. Eğer k değeri 3 olarak belirlenirse algoritma en yakın 3 komşuya bakarak sınıflandırma hakkında karar verir.
- Veri setine katılacak olan yeni test verilerinin mevcut verilere olan uzaklıkları tek tek hesaplanır. Bu durum yukarıda da bahsedildiği gibi büyük veri setlerinde işlem süresini uzatmaktadır.
- İlgili uzaklıklardan en yakın k komşu ele alınır. Öznitelik değerlerine göre k komşu veya komşuların sınıfına atanır.
- Seçilen sınıf, tahmin edilmesi beklenen gözlem değerinin sınıfı olarak kabul edilir. Yani yeni veri etiketlenmiş (label) olur.

Bu tezde indirilmiş olan 3D point cloud nokta kümeleri KNN algoritma çeşitleri, SVM ve SaCR ile karşılaştırılarak başarımlar oranları analiz edilerek, başarımlar oranları arttırılmaya çalışılmıştır. Yapılan çalışmalara ve sonuçlarına ilerleyen bölümlerden ulaşılabilir.

6. DESTEK VEKTÖR MAKİNELERİ İLE LİDAR GÖRÜNTÜLERİNİN SINIFLANDIRILMASI

Matlab sınıflandırma yapmak için birçok fonksiyona sahiptir. Bu bölümde SVM üzerinden yapılan sınıflandırma çalışmalarının sonuçları incelenecektir. Yapılan işlemler birçok sınıf barındıran 2 veri seti (OMA_198_PC3.txt, OMA_198_CLS.txt) üzerinden yapılmıştır.

Eğitim işlemleri Matlab-Classification Learner üzerinden yapılmıştır. Classification Learner ile işlem yapmadan önce kullanılacak olan veri seti kullanıma hazır hale getirilerek, kullanılacak öznitelikler ve sınıflar bir matris içerisinde toplanmıştır. Bundan sonra ise Classification Learner aracına tıklanarak oluşturulan matris seçilmiş ve ardından da kullanılacak olan sınıflandırma yöntemi seçilerek eğitim işlemi tamamlanmıştır. Eğitime başlamadan önce “Use Paralel” kısmı devre dışı bırakılmıştır. Örnek olarak verecek olursak 4 öznitelik için elde edilen Classification Learner sonucuna Şekil 6.1’den ulaşılabilir. Sonuç olarak eğitim dosyası ayrı bir fonksiyon şeklinde çalıştırılmıştır.



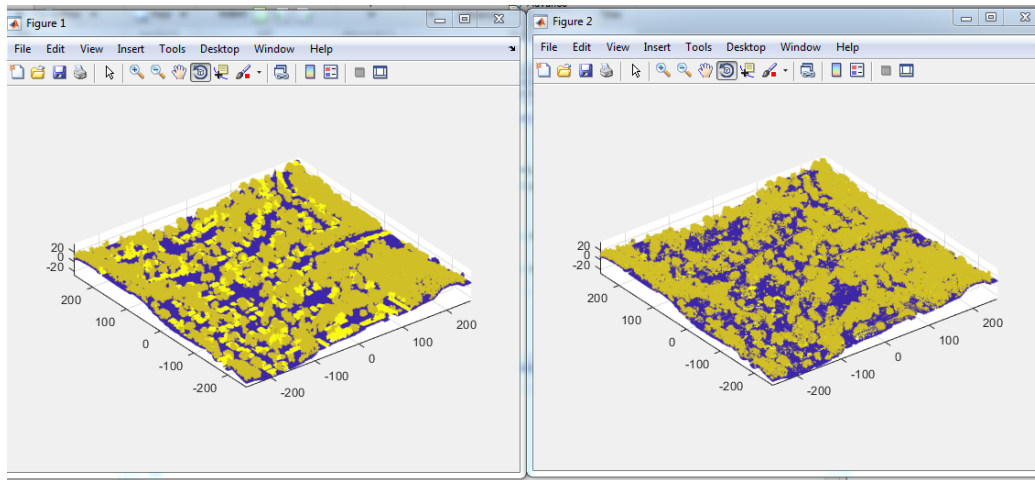
Şekil 6.1: 4 özellik için Classification Learner işlemi sonucu

Kullanılan veri setlerinde 0, 2, 5, ve 6. olmak üzeri 4 adet sınıf içermektedir. Yapılan işlemlerde 0 sınıf etiketine herhangi bir atama yapılmadığı için 0 etiketi atılarak işlemlere o şekilde devam edilmiştir.

Yapılan çalışmalarda çoklu sınıflandırmalar yapılarak, kullanılan öznitelik sayısı artırılmış ve elde edilen başarımlar oranları iyileştirilmeye çalışılmıştır. Yapılan testlere ve sonuçlarına aşağıdan ulaşabilirsiniz.

1-) Z koordinatı, ışıklılık ve sınıf (3 öznitelik) değerleri alınarak işlem yapılmıştır. Elde edilen sonuçlar eğitim ve test için ayrı ayrı incelenmiştir. Elde edilen verilere aşağıdan ulaşılabilir.

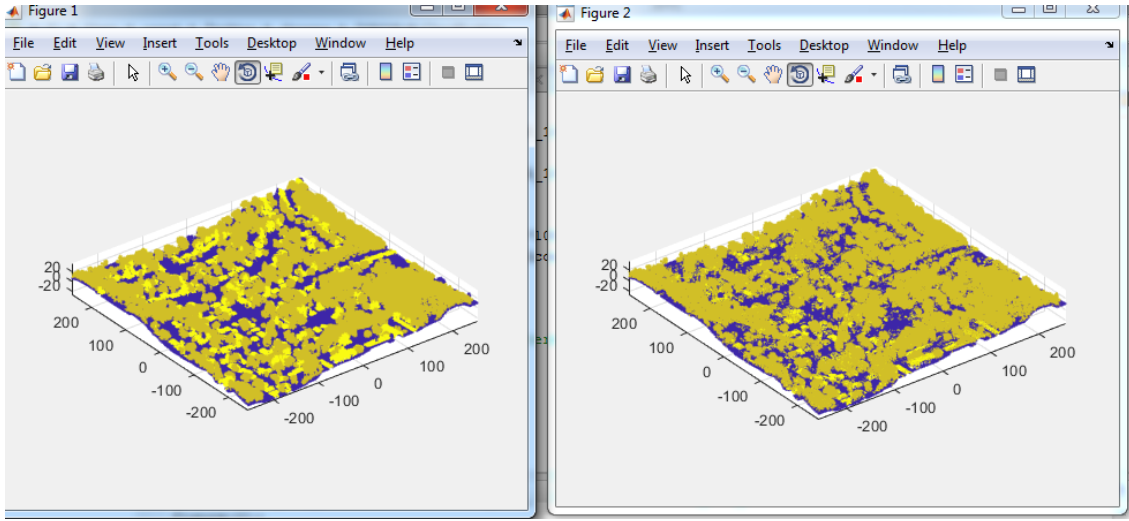
- **Eğitim Kısmı:** Tüm SVM modelleri için uygulanmıştır. En iyi sonuç %87.5 ile gaussian SVM üzerinden alınmıştır.
- **Test Kısmı:** Gaussian SVM fonksiyonu üzerinden yapılmıştır. Accuracy= %79 elde edilmiştir. Test kısmının görsel sonucuna Şekil 6.2'den ulaşılabilir.



Şekil 6.2: 3 sınıf etiketi için elde edilen sonuçlar

2-) Z koordinatı, ışıklılık, dönüş sayısı ve sınıf (4 öznitelik) değerleri alınarak işlem yapılmıştır.

- **Eğitim Kısmı:** Tüm SVM modelleri için uygulanmıştır. En iyi sonuç %86.9 ile Gaussian SVM üzerinden alınmıştır.
- **Test Kısmı:** Gaussian SVM fonksiyonu üzerinden yapılmıştır. Accuracy = %82 elde edilmiştir. Test kısmının görsel sonucuna Şekil 6.3'den ulaşılabilir.



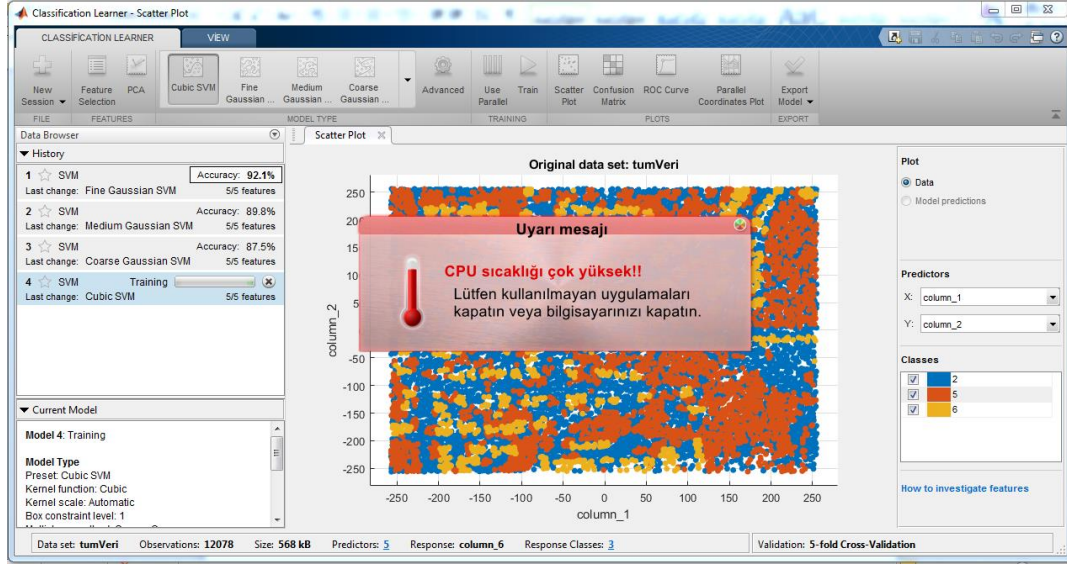
Şekil 6.3: 4 özellik için elde edilen sonuçlar

3-) X, Y, Z koordinatları, ışıklılık, dönüş sayısı ve sınıf (6 öznitelik) değerleri alınarak işlem yapılmıştır.

İşlemler sırasında fazlaca iş yükü olması sebebiyle bilgisayar eğitim sürecini tamamlayamadan kapanmıştır. Bilgisayarın yapılan işlemler sonucunda verdiği uyarıya Şekil 6.4'deki görselden ulaşabilirsiniz.

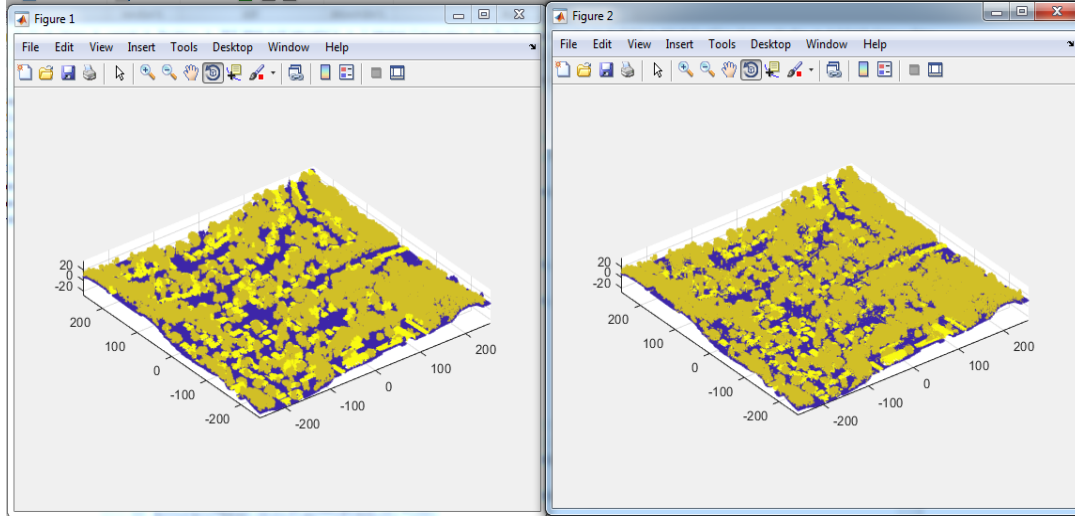
Bilgisayardaki yetersizlikten dolayı veri setinde 1:100 oranında alt örnekleme yapılarak işlemlere devam edilmiştir. İlerleyen bölümlerde yapılan işlemlerde bu aşamaya kadar kullanılan bilgisayar yerine, i7 işlemci ve NVIDIA GEFORCE 960M ekran kartına sahip bir bilgisayar kullanılmıştır.

- **Eğitim Kısmı:** Yapılan çalışmada X ve Y koordinatları da eklenerek eğitimdeki başarımlar oranını gaussian SVM ile %92.1'e çıkmıştır. Eğitim kısmı bilgisayar yetersizliğinden dolayı tam anlamıyla tamamlanamamıştır.



Şekil 6.4: 6 özellik için Classification Learner işlemi sonucu

- **Test Kısmı:** Gaussian SVM ile fonksiyonu üzerinden yapılmıştır. Accuracy = %87 elde edilmiştir. Test kısmının görsel sonucuna Şekil 6.5'den ulaşılabilir.



Şekil 6.5: 6 özellik için elde edilen sonuçlar

7. FARKLI SINIFLARDA ÖZNİTELİK SAYISININDAKİ DEĞİŞİMİN SINIFLANDIRMAYA ETKİSİ

Matlab üzerinden yapılan işlemde JAX_166_CLS ve JAX_166_PC3 veri setlerinde belli örnek sayıları için elde edilen başarımların oranları incelenmiştir. Öznitelik sayısındaki değişimlerin SVM ve KNN sonuçlarını nasıl etkileyeceği gözlemlenmiştir. Bu işlemler öncelikli olarak z koordinatı, ışıklılık ve sınıf değerleri üzerinden yapılmıştır. Öznitelik sayısını arttırarak başarımların artırılması hedeflenmiştir ve bu öznitelikler farklı SVM ve KNN çeşitleri için denenmiştir.

Bu uygulamalar yapılırken test sonuçlarının başarılı bir şekilde gelmesi amacıyla dikkat edilmesi gereken hususlardan bir tanesi SVM yöntemi için “Kernel Scal Mode= Auto” olarak ayarlanmasıdır. Eğer bu şekilde ayarlanma yapılmazsa SVM sonuçlarında elde edilen veriler tutarsızlık gösterecektir. KNN için kernel ayarına ihtiyaç duyulmamaktadır.

Yapılan işlemlerde işlemlerin kolaylığı açısından JAX_166_PC3 dosyasından 1:5 oranında alt örnekleme yapılarak işlemlere o şekilde devam edilmiştir. Normal şartlarda JAX_166_PC3 veri setinde 403780 tane satır ve 6 tane sütün (öznitelik) bulunmaktadır. 1:5 oranında alt örnekleme sonucunda 80756 satır ve 6 sütün üzerinden işlemler yapılmıştır. Bunun haricinde sınıflar dosyasında 0 etiketi kullanılan satırlar etiketsiz olarak isimlendirilmektedir. 0 etiketine sahip değerlerde atıldıktan sonra veri setindeki örnek sayısı 73209’e düşürülmüştür ve JAX_166_PC3 dosyası için işlemler daha rahat, hızlı şekilde gerçekleştirilmiştir. 0 etiketi çıkarıldıktan sonra geriye 2, 5, 6, 9, 17 olmak üzere 5 adet sınıf etiketi kalmıştır.

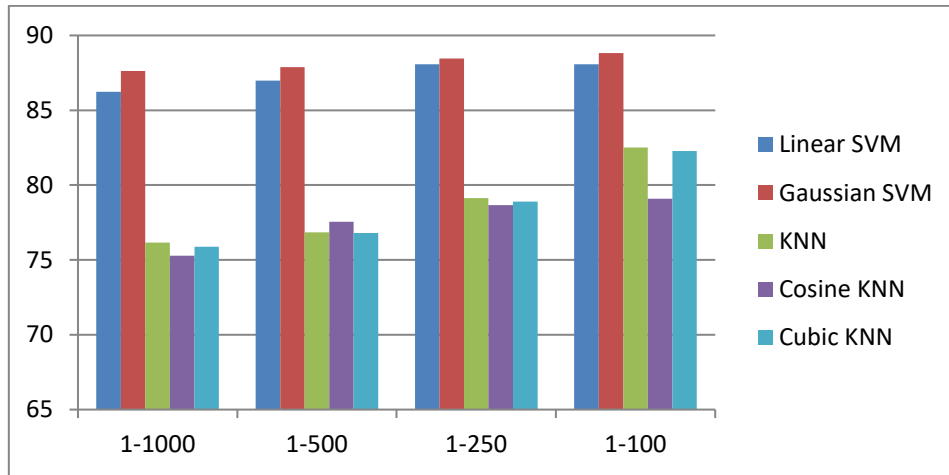
Yapılan testlerle alakalı sonuçlara aşağıdaki tablolardan ulaşılabilir. Ayrıca tabloda ki değerlerin daha basit anlaşılması için tablodaki sonuçlara göre bar grafikleri oluşturulmuştur.

Alt örneklemenin dışında dosyalarda kullanılan TDR oranları 1:100, 1:250, 1:500 ve 1:1000 olarak seçilmiştir. Elde edilen en iyi sonuçlar tablolarda koyu olarak yazılmıştır.

SVM ve KNN, 3 öznitelik için karşılaştırıldığında sonuçlar incelendiği zaman en iyi sonuçlar gaussian SVM’de elde edilmiştir. Elde edilen test sonuçlarına Tablo7.1 ve Şekil 7.1’den ulaşabilirsiniz.

Tablo 7.1: Z, ışıklılık, sınıf'a (3 öznitelik) göre başarımları

TDR	Linear SVM	Gaussian SVM	KNN	Cosine KNN	Cubic KNN
1:1000	86.2285	87.6327	76.1505	75.2817	75.8746
1:500	86.9777	87.8849	76.8395	77.5569	76.8012
1:250	88.0722	88.4531	79.1352	78.6529	78.8996
1:100	88.0689	88.8174	82.5155	79.0793	82.2826

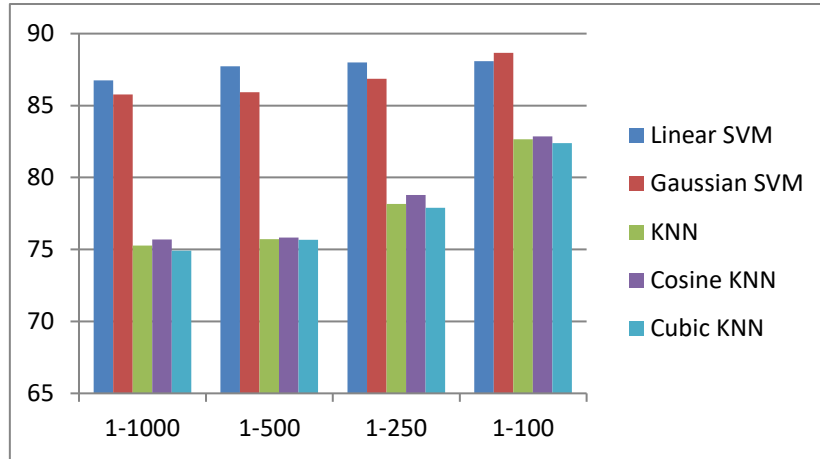


Şekil 7.1: 3 özniteliğe göre elde edilen başarımları

SVM ve KNN, 4 öznitelik için karşılaştırıldığında sonuçlar incelendiği zaman en iyi sonuçlar baskın olarak linear SVM'de elde edilmiştir. Elde edilen test sonuçlarına Tablo7.2 ve Şekil 7.2'den ulaşabilirsiniz.

Tablo 7.2: Z, ışıklılık, dönüş sayısı, sınıf'a (4 öznitelik) göre başarımları

TDR	Linear SVM	Gaussian SVM	KNN	Cosine KNN	Cubic KNN
1: 1000	86.7475	85.7763	75.2626	75.7038	74.9252
1:500	87.7188	85.9240	75.7224	75.8336	75.6705
1:250	87.9859	86.8556	78.159	78.7817	77.8912
1:100	88.0855	88.6726	82.6657	82.8615	82.3956

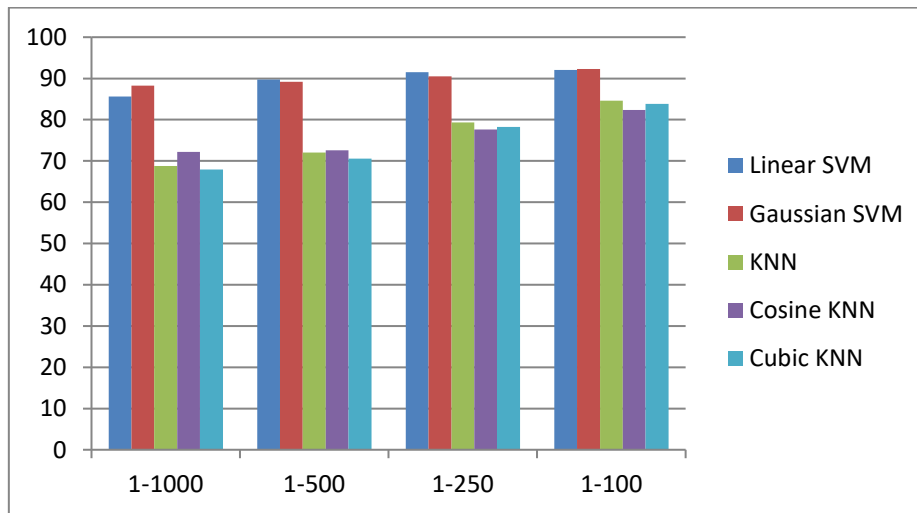


Şekil 7.2: 4 özniteliğe göre elde edilen başarımları

SVM ve KNN, 6 öznitelik için karşılaştırıldığında sonuçlar incelendiği zaman en iyi sonuçlar baskın olarak gaussian SVM’de elde edilmiştir. Elde edilen test sonuçlarına Tablo7.3 ve Şekil 7.3’den ulaşabilirsiniz.

Tablo 7.3: X, Y, Z, ışıklılık, dönüş sayısı, sınıf’a (6 öznitelik) göre başarımları

TDR	Linear SVM	Gaussian SVM	KNN	Cosine KNN	Cubic KNN
1: 1000	85.6493	88.2719	68.7607	72.1578	67.8988
1:500	89.7629	89.1613	72.0056	72.6031	70.5863
1:250	91.5398	90.4712	79.3380	77.6199	78.2597
1:100	92.0786	92.3294	84.6451	82.3970	83.8442



Şekil 7.3: 6 özniteliğe göre elde edilen başarımları

8. SACR YÖNTEMİ VE PARAMETRELERİNİN LİDAR DATASETE GÖRE KONFIGÜRE EDİLMESİ

SaCR yöntemi hiperspektral görüntüler üzerinde sınıflandırma işlemi yapmak için kullanılan yöntemlerden bir tanesidir. Hiperspektral görüntüler genel olarak lidar verilerine göre çok daha büyük boyutlara sahip ve çok daha fazla öznitelik üzerinden sonuçlar elde edilmesini sağlamaktadır. SaCR yöntemi için github [17] sitesinden indirilen kod üzerinden denemeler yapılmış ve elde edilen sonuçlara göre değişkenler analiz edilerek, kod lidar veri seti üzerinde kullanılacak şekilde konfigüre edilmiştir.

Yapılan çalışmalarda eğitimde kullanılan sınıf sayılarında dengesizlik olduğu için test sonuçlarında her sınıf için başarılı sonuç elde edilmemiştir. Class in blance durumu göz ardı edilmiştir.

Ayrıca SaCR yönteminde diğer yöntemlerden farklı olarak uzamsal benzerlikler göz önüne alınmaktadır. Uzamsal benzerlikde komşular ile olan ilişkiler incelenmekte olup, pixel bazlı incelemelerden çıkılarak az da olsa grupsal bazlı incelemeler söz konusu olabilmektedir. Örnek verilecek olursa yangında bir pixel zarar gördüyse komşu pixel de muhtemelen zarar görmüştür yorumu uzamsal incelemelerde yapılabilir.

Hiperspektral görüntüler n tane arka arkaya görüntülerden oluşmaktadır. N adet görüntü n adet öznitelik anlamına gelmektedir. Lidar veri seti için 5 öz nitelikte sınıflandırma yapılırken, hiperspektral görüntüler için 200 tane öznitelik bulunmaktadır. Bundan dolayı hiperspektral görüntülerde test başarımları oranları lidar'a göre daha fazla gelebilmektedir. Kısacası hiperspektral görüntülerde öznitelik sayısı fazlayken, örnek sayısı daha azdır ve lidar veri setinde ise öznitelik sayısı azken, örnek sayısı daha fazladır yorumu yapılabilir.

SaCR yönteminde 3 tane sonuca doğrudan etki etmekte olan parametre bulunmaktadır (c , λ , γ) ve bu parametreler regülasyon parametreleri olarak bilinmektedir. [18] Bu parametreleri lidar veri setine göre konfigüre edilmesi gerekmektedir. Orijinal kodda hiperspektral görüntü üzerinde alınan değerlerde $c=4$, $\lambda=0.01$, $\gamma=1$ olarak alınmıştır. Yapılan değerlerde deneme yanılma yolu ile lidar için en uygun regülasyon parametreleri belirlenmiştir.

Asıl kodda smoothing ve ortalama alma gibi işlemler yapılırken, lidar veri seti için ayarlanan kodda bu işlemler yapılmayarak kod o şekilde konfigürasyon edilmiştir.

Yapılan bu işlemde sınıflandırma başarımlarının SVM ve KNN yöntemine göre artış gözlemlenmesi planlanmıştır. Testler işlem uzunluğu sebebiyle JAX_166_CLS ve JAX_166_PC3 veri setinin 1:200 oranında alt örnekleme yapılmasıyla gerçekleştirilmiştir.

Elde edilen ilk sonuçlarda lambda ve gamma değerleri sabit tutularak sadece c değeri değiştirilerek sonuçlar test edilmiştir ve elde edilen sonuçlara göre en yüksek başarımlar $c=2$ 'de iken alınmıştır. C değişkeninin test sonuçlarına Tablo 8.1'den ulaşılabilir.

Tablo 8.1: Lambda=0.01, gamma=1 iken edilen c değerleri

C Değeri	Başarımlar Oranı
4	0.6110
0	0.5462
1	0.6245
2	0.6403
3	0.6279
2.5	0.6351
2.1	0.6397
2.01	0.6401

Elde edilen ilk sonuçlarda c ve gamma değerleri sabit tutularak sadece lambda değeri değiştirilerek sonuçlar test edilmiştir ve elde edilen sonuçlara göre en yüksek başarımlar $\lambda=10^{-7}$ 'de elde edilmiştir. Lambda değeri 10^{-7} olarak belirlenmiştir. Lambda değişkeninin test sonuçlarına Tablo 8.2'den ulaşabilirsiniz.

Tablo 8.2: C=2, gamma=1 iken elde edilen lambda değerleri

Lambda Değeri	Başarım Oranı
0.01	0.6403
0.1	0.5757
0.05	0.5930
0.02	0.6222
0.015	0.6309
0.09	0.5779
0.9	0.5292
1	0.5272
2	0.5169
5	0.50
0.009	0.6431
0.008	0.6464
0.005	0.6602
0.002	0.6929
$10^{(-5)}$	0.8495
$10^{(-6)}$	0.8536
$10^{(-7)}$	0.8538
$10^{(-8)}$	0.8521
$10^{(-9)}$	0.8519

Elde edilen sonuçlara göre c ve lambda değerleri sabit tutularak sadece gama değeri değiştirilerek sonuçlar test edilmiştir ve elde edilen sonuçlara göre en yüksek başarımlı oranı gamma=0.95’de elde edilmiştir. Gamma değeri için test sonuçlarına Tablo 8.3’den ulaşabilirsiniz.

Tablo 8.3: $C=2$, $\lambda=10^{-7}$ iken elde edilen gamma değerleri

Gamma Değeri	Başarım Oranı
1	0.8537
2	0.8532
5	0.8526
10	0.8522
0.009	0.8494
0.1	0.8536
0.5	0.8537
0.9	0.8537
1.01	0.8537
0.95	0.853861
0.99	0.853806

Yapılan test sonuçlarına göre değişken değerleri $c=2$, $\lambda=10^{-7}$ ve $\gamma=0.95$ olarak elde edilmiştir. Daha sonrasında elde edilen c değeri, daha sağlıklı sonuçlar vermesi için λ ve γ değerine göre tekrar teste sokulmuştur. $\lambda=10^{-7}$, $\gamma=0.95$ iken c değerinin test sonuçlarına Tablo 8.4'den ulaşılabilir.

Tablo 8.4: Lambda= 10^{-7} , gamma=0.95 iken elde edilen c değeri

C Değeri	Başarım Oranı
2	0.853861
1.90	0.853669
1.98	0.853366
1.99	0.853476
2.09	0.853696
2.05	0.853476
2.1	0.853779
2.3	0.854027
2.5	0.853586
2.4	0.853751
2.35	0.853806
2.32	0.853999
2.31	0.853944
2.295	0.853996
2.325	0.853995
10	0.745786

Sonuç olarak yapılan testler ve denemeler sonucunda hiperspektral görüntülerde kullanılan SaCR yönteminin lidar veri seti için kullanılacak parametreleri $c=2.35$, $\lambda=10^{-7}$ ve $\gamma=0.95$ olarak belirlenmiştir. Bu belirlenen parametrelere göre SaCR yöntemi, SVM ve KNN sonuçlarıyla eşit şartlarda testler yapılarak başarımları karşılaştırılacaktır. SaCR yönteminin SVM ve KNN sonuçlarına göre bir artış göstereceği beklenmektedir.

9. SACR, SVM VE KNN DATASETLERİNİN EŞİT ŞARTLAR ALTINDA FARKLI DATASETLERDE SINIFLANDIRMA TEST SONUÇLARI

Eski yapılan işlemler SVM ve KNN için özellik ve eğitim de kullanılan veri miktarlarındaki değişim oranlarına göre JAX166 veri seti üzerinden karşılaştırmalar yapılmıştır. Bu sefer hiperspektral görüntülerde kullanılan SaCR yönteminin SVM ve KNN için karşılaştırması JAX166, OMA198 ve JAX280 olmak üzere 3 ayrı veri setleri üzerinden yapılmaktadır. Yapılan işlemlerde karşılaştırma sonuçlarının verimliliği açısından dosya bazında SaCR, SVM ve KNN için aynı şartlarda sonuçlar test edilmiştir.

Dosyalardaki veri miktarlarının farklı olması ve bilgisayar yeterliliği baz alındığı zaman dosyalara maksimum verimlilik sağlanacak alt örneklemeler yapılarak işlemler yapılmıştır. Buna bağlı olarak JAX166 dosyası 1:5 oranında, OMA198 dosyası 1:20 oranında ve JAX280 dosyası 1:10 oranında alt örneklemelere tabi tutularak işlemlere devam edilmiştir. Tablolarda geri kalan örnekleme oranları eğitimde kullanılacak veri miktarını göstermekte olup eğitimde kullanılan verilerin haricinde kalan veriler test için kullanılmıştır.

Yukarıdaki koşullara göre yapılan işlemlerde JAX166 dosyası normal şartlarda 403780 tane satır ve 6 tane sütun(öznetelik) içermektedir. 1:5 oranında yapılan alt örnekleme sonucunda satır sayısı 80786'ya düşürülmüştür. Ayrıca bu dosya 0, 2, 5, 6, 17 olmak üzere toplamda 5 adet sınıf barındırmaktadır. Bu sınıflardan 0 etiketsiz sınıfları ifade etmekte olduğu için atılmıştır. 0 sınıf etiketli satırlar atıldığı zaman elde işlem yapılabilecek 73283 adet satır kalmakta ve JAX166 veri seti için karşılaştırma yapılırken en son 73283 satır ve 6 sütun üzerinden işlemler yapılmıştır. JAX166 verisinin test başarımlarına Tablo 9.1'den ulaşılabilir.

OMA198 veri seti için toplamda 1288419 adet satır mevcuttur ve işlemlerin kolaylığı açısından bu veri seti üzerinden yapılan işlemleri yapmadan önce 1:20 oranında alt örnekleme yapılarak veri 64421 satıra örneklenmiştir. Elde edilen bu oranda 0 sınıf etiketlerinin atılmasıyla birlikte elde 60983 satır ve 6 sütun (özellik)'den oluşan veri kalmıştır ve OMA198 için işlemler bu veri üzerinden yapılmıştır. Ayrıca OMA198 veri seti 0, 2, 5, 6 olmak üzere 4 sınıf barındırmakta olup yapılan işlemler 0 sınıf etiketi atılarak 2, 5, 6 sınıfları üzerinden yapılmıştır. OMA198 verisinin test başarımlarına Tablo 9.2'den ulaşılabilir.

JAX280 veri seti için toplamda 772840 adet satır mevcuttur ve işlemlerin kolaylığı açısından bu veri seti üzerinden yapılan işlemleri yapmadan önce 1:10 oranında alt örnekleme yapılarak veri 77284 satıra örneklenmiştir. Elde edilen bu oranda 0 sınıf etiketlerinin atılmasıyla birlikte elde 72630 satır ve 6 sütun (özellik)’den oluşan veri kalmıştır ve JAX280 için işlemler bu veri üzerinden yapılmıştır. Ayrıca JAX280 veri seti 0, 2, 5, 6 olmak üzere 4 sınıf barındırmakta olup yapılan işlemler 0 sınıf etiketi atılarak 2, 5, 6 sınıfları üzerinden yapılmıştır. JAX280 verisinin test başarımlarına Tablo 9.3’den ulaşılabilir.

Yapılan işlemlerde TDR oranları 1:1000, 1:500, 1:250, 1:150, 1:100, 1:50 olarak seçilmiştir. 3 farklı veri seti için elde edilen sonuçlara aşağıdaki tablolardan ulaşılabilir. Ayrıca sonuçların ortalamalarının bar grafiğine de Şekil 9.5’den ulaşılabilir.

Tablo 9.1: JAX166 veri seti için elde edilen sonuçlar

TDR	Linear SVM	Gaussian SVM	KNN	Cosine KNN	Cubic KNN	SaCR
1:1000	85.6493	80.0598	68.7608	72.1578	67.8988	80.0598
1:500	89.1613	82.0417	72.0056	72.6031	70.5863	82.0417
1:250	90.4849	90.2574	79.3380	77.6199	78.2597	90.7577
1:150	91.8804	90.9045	81.7114	79.3637	80.6687	93.0561
1:100	92.1365	92.4342	84.6451	82.3970	84.8442	92.4167
1:50	93.1462	93.9986	87.4695	85.0258	86.9154	95.2154

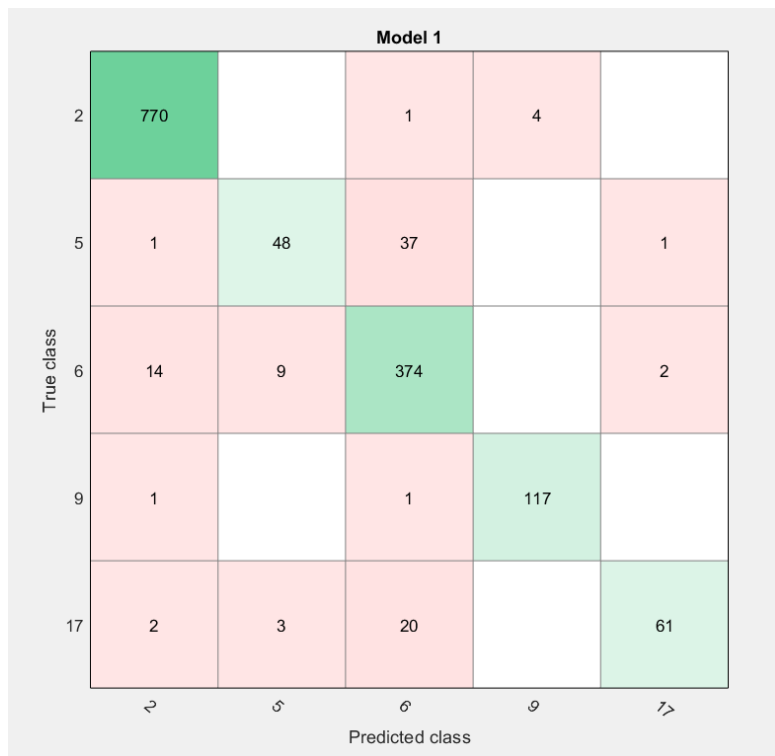
JAX166 veri setinin tüm sınıflandırma çeşitleri için görselleştirilmeleri yapılmıştır. Yapılan çalışmada SVM ile alakalı görselleştirmelere Şekil 9.1’den, KNN ile alakalı görselleştirmelere Şekil 9.2’den ve SaCR ile ilgili görselleştirmelere Şekil 9.3’den ulaşılabilir.

Tablo 9.1’de elde edilen sonuçlara göre JAX166 veri seti için en iyi sonuçlar linear SVM ve SaCR yöntemlerinde elde edilmiştir. Elde edilen sonuçlarda 1:250’den sonraki adım aralıklarında eğitimde kullanılan veri miktarının az olması sebebiyle linear SVM’in SaCR’ye göre daha başarılı sonuçlar verdiği gözlemlenmiştir.

JAX166 veri seti için eğitim ve testte kullanılan ham veri miktarı 73283 iken bu veri 1:1000 adım aralığı ile eğitildiğinde 73 veri eğitim için kullanılırken, geriye kalan 73210 ise test için

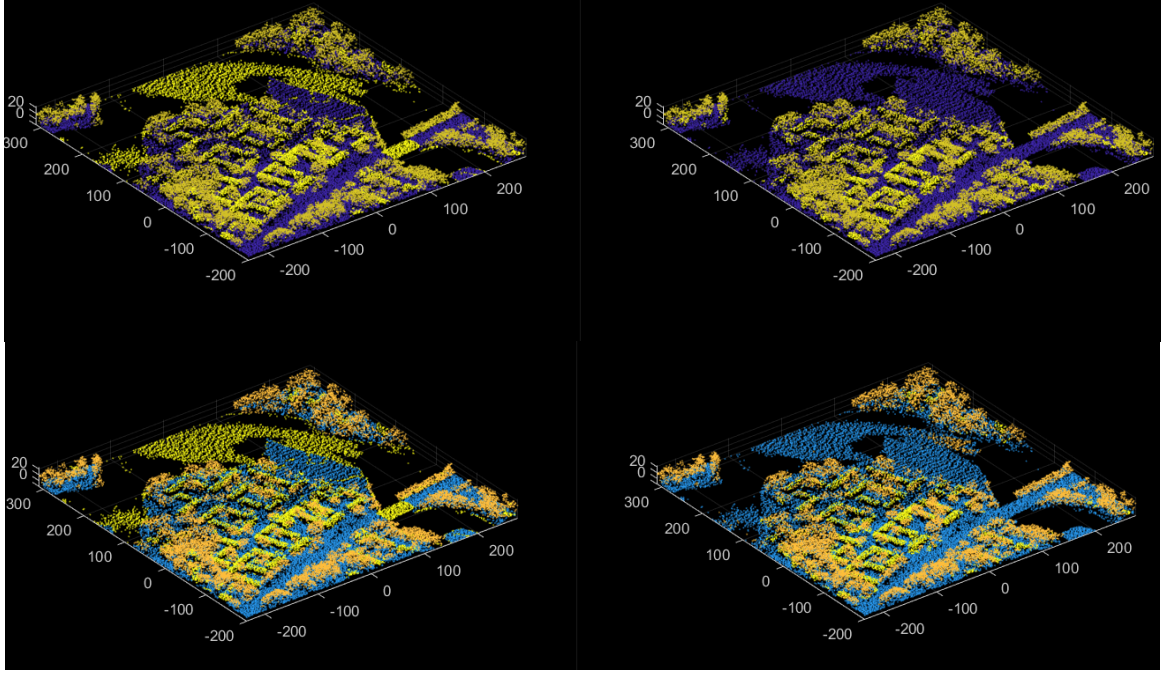
kullanılmaktadır. Bu durumda SaCR'nin test başarımlarının düşmesindeki en etkin sebebin, eğitim veri miktarındaki yetersizlik olduğu sonucuna varılır. Bu durumda gerçek hayattaki uygulamalarda sınıflandırma için kullanılacak yöntemlere göre en yüksek başarımlar elde edilecek test veri miktarı optimal seviyede belirlenmelidir.

JAX166 veri seti için gaussian SVM ile elde edilen eğitim sonucunda %93.5'lik başarımlar elde edilmiştir. Elde edilen eğitim sonucuna göre “confusionmatrix” oluşturulmuştur. Bu matris sayesinde sınıflar için elde edilen doğru ve yanlış tahminler açık bir şekilde gösterilmektedir. 5 ayrı sınıf için elde edilen confusionmatrix'e Şekil 9.4'deki görselden ulaşılabilir.

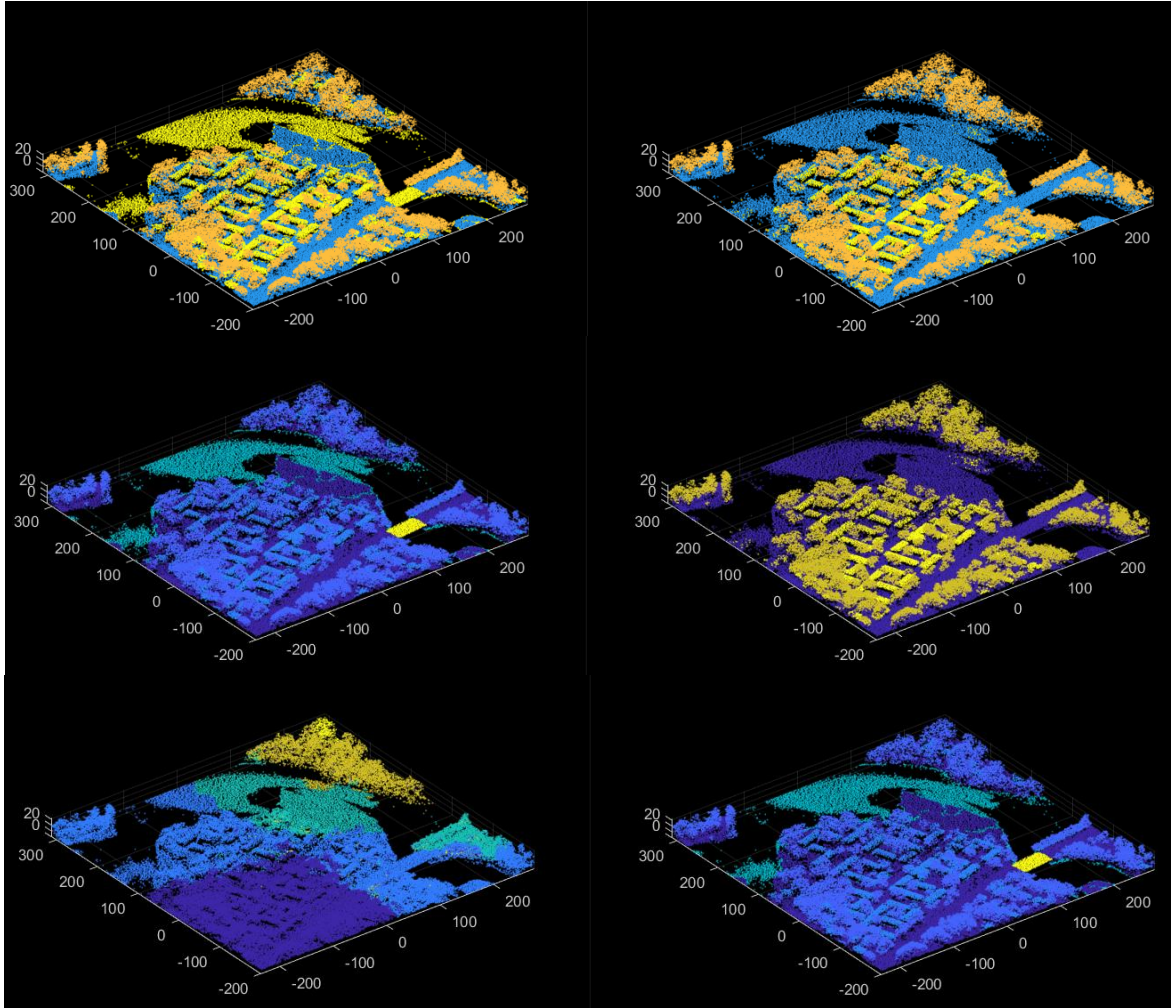


Şekil 9.4: JAX166 veri seti için elde edilen confusionmatrix sonucu

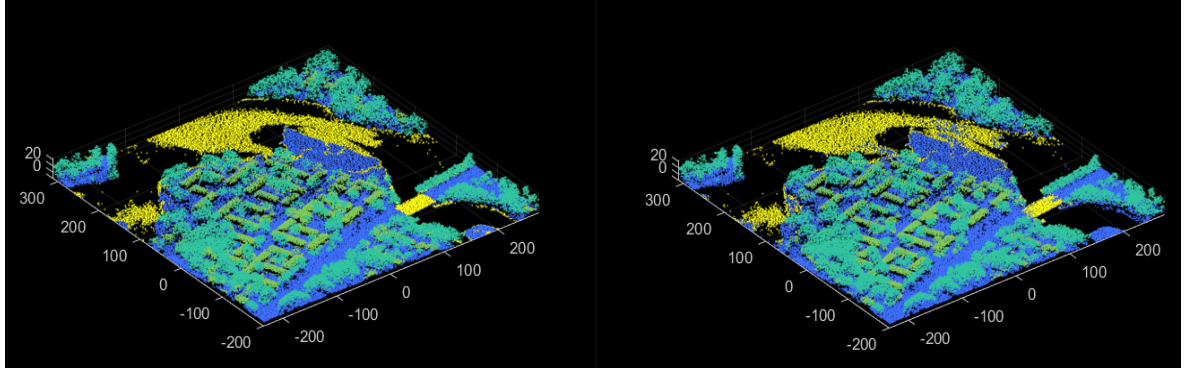
Confusionmatrix incelendiği zaman JAX166 veri seti için 0 haricinde 5 ayrı sınıf olduğu görülmektedir. Bu matris sayesinde sınıfların kaç tanesinin doğru, kaç tanesinin yanlış tahmin edildiği açıkça görülmektedir. Örnek verilecek olursa toplam 788 adet 2 numaralı sınıf etiketi bulunmaktadır ve 770 tane doğru tahmin yapılırken 18 adet yanlış tahmin işlemi yapılmıştır. 2 yerine; 1 adet 5, 14 adet 6, 1 adet 9, 2 adette 17 sınıfı tahmin edilmiştir. Kısacası confusionmatrix ile sınıfların tahminleri hakkında detaylı bilgiye basit bir şekilde ulaşma imkanımız artmaktadır.



Şekil9.1: JAX166 veri setinin SVM görselleştirme sonuçları



Şekil 9.2: JAX166 veri setinin KNN görselleştirme sonuçları



Şekil 9.3: JAX166 veri setinin SaCR görselleştirme sonuçları

Tablo 9.2: OMA198 veri seti için elde edilen sonuçlar

TDR	Linear SVM	Gaussian SVM	KNN	Cosine KNN	Cubic KNN	SaCR
1:1000	84.1265	84.6586	80.4540	81.3190	80.5427	77.8789
1:500	87.1323	86.6079	85.0868	84.8806	84.5654	86.5067
1:250	87.0349	86.8634	85.3972	84.9716	85.1661	89.4640
1:150	87.1399	88.3940	86.0682	85.6197	85.9165	91.4462
1:100	87.2023	88.2561	86.9229	86.5449	86.6503	93.4648
1:50	87.4045	88.9236	87.9655	87.6022	87.8219	94.8091

Tablo 9.2’de elde edilen sonuçlara göre OMA198 veri seti için en iyi sonuçlar linear SVM, gaussian SVM ve SaCR yöntemlerinde elde edilmiştir. Elde edilen sonuçlarda 1:250’den sonraki adım aralıklarında eğitimde kullanılan veri miktarının az olması sebebiyle linear ve gaussian SVM’in SaCR’ye göre daha başarılı sonuçlar verdiği gözlemlenmiştir.

OMA198 veri seti için eğitim ve testte kullanılan ham veri miktarı 60983 iken bu veri 1:1000 adım aralığı ile eğitildiğinde 60 veri eğitim için kullanılırken, geriye kalan 60923 ise test için kullanılmaktadır. Bu durumda SaCR’nin test başarımlarının düşmesindeki en etkin sebebin, JAX166 veri setinde de olduğu gibi eğitim veri miktarındaki yetersizlik olduğu sonucuna varılır. Ayrıca Tablo 9.1 ve Tablo 9.2 incelendiği zaman SVM’in KNN’e başarımlarının daha yüksek olduğu görülmektedir. JAX166 ve OMA198 veri setleri için 2 durumda da SVM başarımlarını KNN’e göre daha fazla olduğu gözlemlenmiştir.

Tablo 9.3. JAX280 veri seti için elde edilen sonuçlar

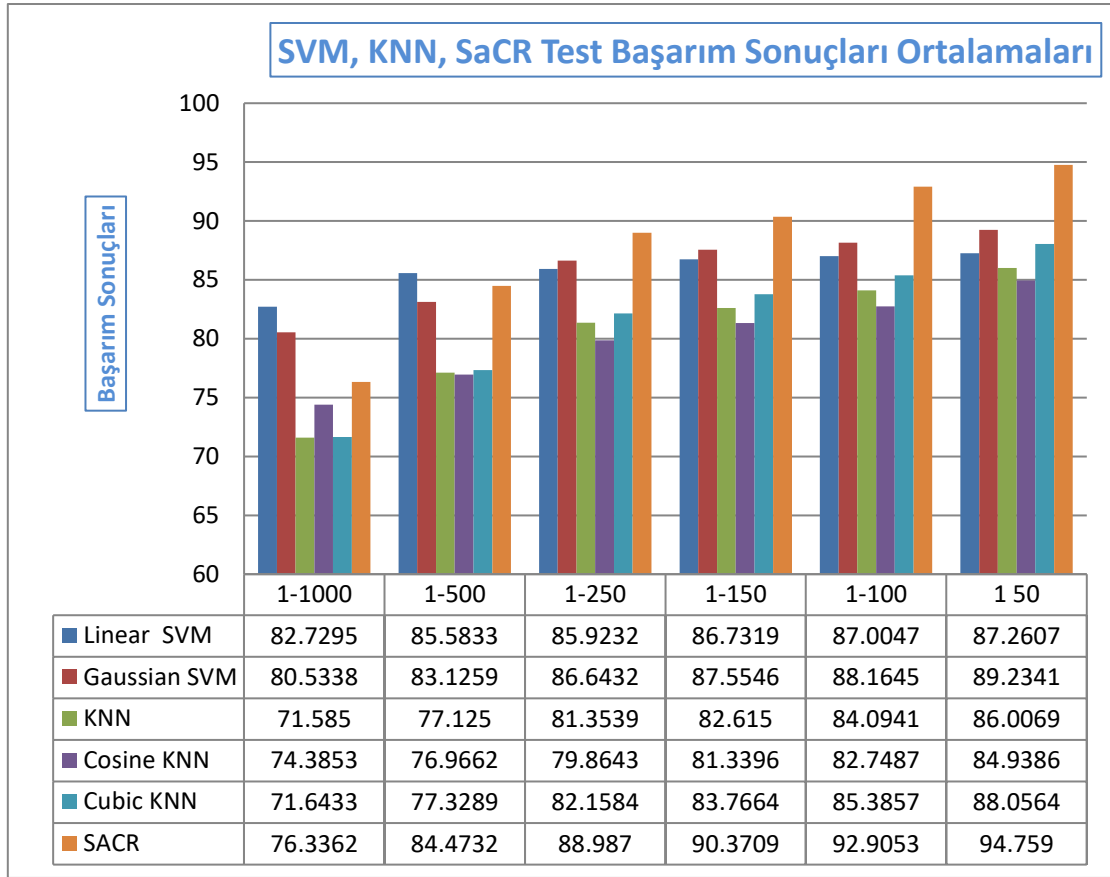
TDR	Linear	Gaussian	KNN	Cosine	Cubic	SaCR
	SVM	SVM		KNN	KNN	
1:1000	78.4128	76.8830	65.5402	69.6790	66.4884	71.0699
1:500	80.4564	80.7282	74.2826	73.4148	76.8349	84.8711
1:250	80.2499	82.8087	79.3265	77.0013	83.0493	87.2392
1:150	81.1754	83.3654	80.0654	79.0353	84.7141	86.6103
1:100	81.6754	83.8032	80.7143	79.3041	84.6627	92.8345
1:50	81.2313	84.7802	82.5857	82.1878	89.4320	94.2495

Tablo 9.3’de elde edilen sonuçlara göre JAX280 veri seti için en iyi sonuçlar linear SVM ve SaCR yöntemlerinde elde edilmiştir. Elde edilen sonuçlarda Tablo 1 ve Tablo 2’den farklı olarak SaCR sadece 1:1000 adım aralığında daha az başarımlı gözlemlenmiştir. 1:1000 adım aralığında ise başarımlı oranı en yüksek olan linear SVM olarak görülmektedir.

JAX280 veri seti için eğitim ve testte kullanılan ham veri miktarı 72630 iken bu veri 1:1000 adım aralığı ile eğitildiğinde 72 veri eğitim için kullanılırken, geriye kalan 72558 ise test için kullanılmaktadır. Bu durumda SaCR’nin test başarımlı oranının düşmesindeki en etkin sebebin, JAX166 ve OMA198 veri setlerinde de olduğu gibi eğitim veri miktarındaki yetersizlik olduğu sonucuna varılır.

Sonuç olarak 3 tabloda incelendiği zaman eğitim veri miktarındaki yeterlilik durumunda SaCR yöntemi SVM ve KNN yöntemlerine göre 3 ayrı veri seti üzerinde daha sağlıklı test sonuçları vermiştir.

Ayrıca 3 tabloda elde edilen sonuçlara göre 3 tablonun ortalamaları alınarak bar grafiği oluşturulmuştur. 3 ayrı veri seti için yapılan sınıflandırmaların sonuçlarının ortalaması alınarak hazırlanan bar grafiği incelendiği zaman, düşük adım aralıklarında SaCR yönteminin SVM ve KNN’e göre daha sağlıklı sonuçlar verdiği görülmektedir. Ortalama bar grafiğine Şekil 9.5’den ulaşılabilir.



Şekil 9.5: SVM, KNN ve SaCR test başarım sonuçları ortalaması bar grafiği

SONUÇLAR VE ÖNERİLER

Yapılan bu tez kapsamında belirlenen 3 Boyutlu Nokta Bulutu yapılar makine öğrenmesi kullanılarak sınıflandırılmıştır. İlk olarak veri setleri indirilmiş ve indirilen veri setleri üzerinden görselleştirilmeler, Matlab yazılımı üzerinden yapılmıştır. İndirilen veri setleri sayısal veri setleri olup, sınıflandırmalar ve görselleştirmeler bu veriler üzerinden yapılmıştır. Veri setlerinin görselleştirilmesine Z koordinatı ve ışıklılık değerlerinden başlanmıştır.

Verilerin anlaşılması için Lidar veri seti hakkında detaylı aramalar yapılmıştır. Yapılan araştırmaların ardından SVM sınıflandırma yöntemi ile sınıflandırma işlemlerine başlanmıştır ve aynı koşullar altında aynı veri setleri ile KNN ile sınıflandırma işlemleri yapılarak sınıflandırma başarımları karşılaştırılmıştır.

Bundan sonraki aşamada Linux-Ubuntu işletim sisteminde Tensorflow yazılımı ile pointnet yöntemi kullanılarak sınıflandırma yapılmaya çalışılmıştır. Tensorflow GPU Ubuntu'ya başarıyla kurularak, obje tespiti gibi makine öğrenmesi uygulamaları yapılmıştır ama Lidar data seti pointnet yöntemi ile çalıştırılamamıştır.

Daha sonrasında Matlab yazılımı üzerinden SVM ve KNN için farklı öz niteliklerde sonuçlar analiz edilmiştir. Yapılan işlemlerin ardından hiperspektral görüntülerde kullanılan JSaCR yöntemi lidar veri setine entegre edilmeye çalışılmıştır.

Ortalama alma işlemi tam olarak başaramadığı için JSaCR yöntemi yerine SaCR yöntemi lidar veri setine entegre edilmiştir ve sonuçlar SVM ile KNN'de ki şartlarda çalıştırılarak analizlere devam edilmiş ve başarımları arttırılmıştır.

Ekstra olarak şunu da eklemem gerekir ki makine öğrenmesi, sınıflandırma gibi işlemlerin başarılı bir şekilde yapılabilmesi için kaliteli GPU'ya sahip bilgisayarların tercih edilmesi işlem kolaylığı ve sağlıklı sonuçlar elde edilmesi amacıyla çok önemlidir. Yapılan tezde testlerin büyük bir kısmı bilgisayar yetersizliğinden dolayı farklı bilgisayarlar üzerinden yapılmak zorunda kalmıştır.

Testler sonucunda başarımlar oranları SVM için %94'lere çıkarılırken, KNN için %89'lara ve SaCR içinse %96'lara çıkarılmıştır. Testlerdeki başarımlar oranlarının artırılması için Matlab yazılımında JSaCR yöntemi ve Tensorflow GPU yazılımında pointnet yöntemi denenebilir.

Bu tez kapsamında elde edilen veriler ile ulusal bir makale yayını çıkarılması hedeflenmekte olup eksik yöntemlerin tamamlanması ile birlikte uluslararası bir makale yayınlanması düşünülmektedir.

KAYNAKLAR

- [1] <http://www.grss-ieee.org/community/technical-committees/data-fusion/2019-ieee-grss-data-fusion-contest-data/>
- [2] <https://fenbildergi.aku.edu.tr/wp-content/uploads/2016/12/035506-679-692.pdf>
Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi
- [3] <https://www.autopilotreview.com/experience-waymo-self-driving-car-360-degree-video/>
- [4] <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/what-is-lidar-data-.htm>
- [5] <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/lidar-point-classification.htm>
- [6] <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/what-is-intensity-data-.htm>
- [7] <https://medium.com/deep-learning-turkiye/makine-%C3%B6%C4%9Frenmesi-e%C4%9Flencelidir-b9d50aad3a62>
- [8] <https://www.emrealadag.com/makine-ogrenmesi-nedir.html>
- [9] <https://www.kizgibikodla.com>
- [10] <https://makineogrenimi.wordpress.com/2017/05/25/makine-ogrenmesi-teknikleri/>
- [11] <https://medium.com/dataseries/7-steps-to-machine-learning-how-to-prepare-for-an-automated-future-78c7918cb35d>
- [12] <https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-4-destek-vekt%C3%B6r-makineleri-2f8010824054>
- [13] <https://fderyckel.github.io/machinelearningwithr/svm.html>
- [14] <https://www.mathworks.com/discovery/support-vector-machine.html:.html>
- [15] <https://medium.com/@ekrem.hatipoglu/machine-learning-classification-support-vector-machine-kernel-trick-part-10-7ab928333158>
- [16] <https://medium.com/@k.ulgen90/makine-%C3%B6%C4%9Frenimi-b%C3%B6l%C3%BCm-2-6d6d120a18e1>
- [17] <https://github.com/junjun-jiang/JSaCR>
- [18] Junjun Jiang, Member, IEEE, Chen Chen, Member, IEEE, Yi Yu, Xinwei Jiang, and Jiayi Ma Member, IEEE “Spatial-Aware Collaborative Representation for Hyperspectral Remote Sensing Image Classification”, IEEE Trans. Geosci. Remote Sensing.

ÖZGEÇMİŞ

Ad-Soyad: Samet ARSLANTÜRK

Doğum Tarihi ve Yeri: 03.06.1995 / ZONGULDAK

E-posta: smtarslanturk@gmail.com

ÖĞRENİM DURUMU:

- **Lise:** 2009-2013, Zonguldak Atatürk Anadolu Lisesi
- **Lisans:** 2014-2020, Kocaeli Üniversitesi Elektronik Ve Haberleşme Mühendisliği (%30 İngilizce)

MESLEKİ DENEYİMLER:

- FEMSAN Elektirik Motorları – Zorunlu Staj 1 (12.06.2017 - 07.07.2017) İstanbul/TÜRKİYE
- POSCOASSAN TST – Proje Memuru (16.11.2017 - 15.05.2018) Kocaeli/TÜRKİYE
- Eagles Beachwear – Work And Travel Programı Kapsamında (15.06.2018 - 16.09.2018) South Caroline/AMERİKA
- NETAS - Long-Term Intern (18.02.2019 - 31.05.2019) İstanbul/TÜRKİYE
- ASELSAN – Zorunlu Staj 2 (01.07.2019 – 02.07.2019) Ankara/TÜRKİYE

KULÜP VE ÜYELİKLER:

- KOU IEEE:
 - Genel Üye (2014-2015)
- KOU Kariyer Ve Danışmanlık Kulübü:
 - Canlı Mülakat Etkinlik Koordinatörü (2015-2016)
- KOU Robotik Kulübü:
 - Kurucu Üye ve Etkinlik Koordinatörü (2017-2018)
- KOU Mühendis Beyinler Kulübü:
 - Kurucu Başkan Yardımcısı ve Akademi Çayı Koordinatörü (2017-2018)
 - Denetleme Kurulu Başkanı (2019-2020)