



T.C.
SELÇUK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



GÖMÜLÜ SİSTEM ÜZERİNDE MAKİNE
ÖĞRENMESİ KULLANARAK YERE NÜFUZ
EDEN RADAR VERİSİNDEN NESNE TESPİTİ

Hayri KILIÇ

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Eylül-2023
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Hayri KILIÇ tarafından hazırlanan “Gömülü Sistem Üzerinde Makine Öğrenmesi Kullanarak Yere Nüfuz Eden Radar Verisinden Nesne Tespiti” adlı tez çalışması 28/09/2023 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof. Dr. Hasan Erdinç KOÇER

.....

Danışman

Prof.Dr. Hasan Erdinç KOÇER

.....

Üye

Doç. Dr. Mehmet Akif ŞAHMAN

.....

Üye

Dr. Öğr. Üyesi Sabri ALTUNKAYA

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Ömer Faruk YÜKSEL
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Hayri KILIÇ

Tarih: 15.09.2023

ÖZET

YÜKSEK LİSANS TEZİ

GÖMÜLÜ SİSTEM ÜZERİNDE MAKİNE ÖĞRENMESİ KULLANARAK YERE NÜFUZ EDEN RADAR VERİSİNDEN NESNE TESPİTİ

Hayri KILIÇ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Hasan Erdiñ KOÇER

2023, 75 Sayfa

Jüri

Prof. Dr. Hasan Erdiñ KOÇER
Doç. Dr. Mehmet Akif ŞAHMAN
Dr. Öğr. Üyesi Sabri ALTUNKAYA

Günümüzde yere nüfuz eden radar (YNR) sistemleri yer altında bulunan nesnelerin tespiti için en popüler çözümdür. Elektromanyetik bir enerjinin yer altına gönderilmesi ve yer altında bulunan hedeflerden gelen yansımaların sürekli olarak alınması prensibine dayanan bu tespit işleminde farklı nesnelerden yansımalar ve enerjinin dağılması gibi yanıltıcı birçok etken bulunmaktadır. Bu etkileri ortadan kaldırmak veya en aza indirmek için çeşitli teknikler geliştirilmeye çalışılmaktadır.

Bu çalışmada makine öğrenmesi yöntemlerinden Yapay Sinir Ağları (YSA) ve Derin Öğrenme modelleri kullanılarak radar verisinden nesne tespit edilmesi hedeflenmiştir. Bu amaçla geliştirilen model üç farklı gömülü kartlı sistem üzerinde uygulanmış ve başarılı sonuçlar elde edilmiştir.

Anahtar Kelimeler: Nesne Tespiti, Yere Nüfuz Eden Radar, Makine Öğrenmesi

ABSTRACT**MS THESIS****Object Detection from Ground Penetrating Radar Data Using
Multilayer Artificial Neural Networks in Embedded Systems****Hayri KILIÇ**

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
SELÇUK UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE / DOCTOR OF PHILOSOPHY
IN MECHANICAL ENGINEERING**

Advisor: Prof. Dr. Hasan Erdiñ KOÇER**2023, 75 Pages****Jury**

**Prof. Dr. Hasan Erdiñ KOÇER
Doç. Dr. Mehmet Akif ŞAHMAN
Dr. Öğr. Üyesi Sabri ALTUNKAYA**

Today, ground penetrating radar (GPR) systems are the most popular solution for detecting underground objects. In this detection process, which is based on the principle of sending electromagnetic energy underground and continuously receiving reflections from underground targets, there are many misleading factors such as reflections from different objects and dispersion of energy. Various techniques are being developed to eliminate or minimize these effects.

In this study, it is aimed to detect objects from radar data by using Artificial Neural Networks (ANN) and Deep Learning models, which are machine learning methods. The model developed for this purpose was implemented on three different embedded card system and successful results were obtained.

Keywords: Object Detection, Ground Penetrating Radar, Machine Learning

ÖNSÖZ

Çalışmalarım esnasında akademik bilgi ve tecrübelerinden fazlasıyla istifade ettiğim, benden hiçbir zaman ilgi ve alakasını esirgemeyen, benim için bir danışmandan daha fazlası olan saygıdeğer tez danışmanım Prof. Dr. Hasan Erdinç KOÇER'e teşekkürlerimi sunarım.

Çalışmamın başından sonuna kadar benden maddi manevi desteklerini esirgemeyen, üzerimdeki haklarını asla ödeyemeyeceğim anne ve babama ayrıca çalışmalarında bana her zaman destek olan kardeşim Enes KILIÇ'a, bu süreçte sabırla bana destek olan sevgili nişanlım Zeynep Buse GÖKÇINAR'a, değerli bilgilerini ve tecrübesini benimle paylaşan ağabeyim gibi sevdiğim Çağatay CANGÜLOĞLU'na ve şu anda çalışmakta olduğum ailem gibi gördüğüm upu.io çalışanlarına teşekkür ederim.

Hayri KILIÇ
KONYA-2023

İÇİNDEKİLER

ÖZET	1
ABSTRACT.....	2
ÖNSÖZ	3
İÇİNDEKİLER	4
SİMGELER VE KISALTMALAR.....	6
1. GİRİŞ.....	8
2. KAYNAK ARAŞTIRMASI	10
3. YERE NÜFUZ EDEN RADAR	15
3.1. Yere Nüfuz Eden Radarlara Genel Bakış	15
3.2. Yere Nüfuz Eden Radarların Çalışma Prensipleri.....	15
3.2.1. Yere Nüfuz Eden Radar Sistemlerinde Karşılaşılan Kayıplar	19
3.2.1.1. Yol Kayıpları	19
3.2.1.2. Hedef Yansıması	20
3.2.1.3. Dağınıklık	20
3.3. Yere Nüfuz Eden Radar Sistemlerinde Kullanılan Teknikler	20
3.3.1. Frekans Adımlı Sürekli Dalga Radar	20
3.3.2. Darbeli Yere Nüfuz Eden Radar	21
3.3.3. İleri Bakan Yere Nüfuz Eden Radar	21
3.4. Yere Nüfuz Eden Radar Veri Tipleri	22
3.4.1. A-Tarama.....	22
3.4.2. B-Tarama.....	22
3.4.3. C-Tarama.....	23
3.5. Yere Nüfuz Eden Radarın Bilimsel Gösterimi ve Veri Seti	24
3.5.1. GprMax	24
3.5.2. Paraview Programı.....	25
4. GÖMÜLÜ SİSTEMLER.....	27
4.1. STM32F407VGT6.....	27
4.2. ESP32-WROOM-32DA.....	29
4.3. NVIDIA Jetson Nano.....	30
5. GÖMÜLÜ SİSTEM ÜZERİNDE NESNE TESPİTİ	32
5.1. Derin Öğrenmeye Genel Bakış.....	33
5.2. Yapay Sinir Ağları.....	33
5.3. Derin Öğrenme Algoritmaları.....	34
5.3.1. Evrimsel Sinir Ağları.....	35
5.3.2. Tekrarlayan Sinir Ağları.....	36
5.4. GprMax ile Yere Nüfuz Eden Radar Verisinin Elde Edilmesi	37

5.5. GprMax Çıktısının Paraview Programı Üzerinde Görüntülenmesi.....	44
5.6. MATLAB Üzerinden Bilgisayar Tabanlı YNR ile Nesne Tespiti	48
5.7. Python ile Elde Edilen Verilerin Eğitimi	51
5.7.1. Eğitim Çıktısını Gömülü Sistem İçin Almak	52
5.8. Gömülü Sistem Üzerinde Test Uygulamaları	53
5.8.1. STM32F407VGT6 ile A-Tarama Verilerinin İşlenmesi.....	53
5.8.2. ESP32-WROOM-32DA ile B-Tarama Verilerinin İşlenmesi.....	57
5.8.3. NVIDIA Jetson Nano ile B-Tarama Verilerinin İşlenmesi.....	64
5.9. Uygulama Sonuçları	65
6. SONUÇLAR VE ÖNERİLER	67
6.1 Sonuçlar	67
6.2 Öneriler	68
KAYNAKLAR	69
ÖZGEÇMİŞ.....	75

SİMGELER VE KISALTMALAR

Simgeler

B: Manyetik yoğunluk
 c: Işık Hızı
 D: Elektrik akı yoğunluğu
 ϵ_c : Kompleks izin verme
 ϵ_r : Relatif dielektrik sabit
 ϵ_ω : Dielektrik malzemelerin frekansa bağlı dielektrik sabiti
 ϵ_0 : Dielektrik sabiti
 ϵ_∞ : Dielektrik sabitlerin yüksek frekansta yaklaşık değeri
 $I_s(t)$: İletken telin taşıdığı akımın y eksenini boyunca yoğunluğu
 J_c : Serbest yüzey akım yoğunluğu
 J_s : İletkenlik akım yoğunluğu
 L_a : Malzemenin Zayıflama Kaybı (dB)
 L_e : Anten Etkinlik Kayıpları (dB)
 L_m : Anten Uyum Kayıpları (dB)
 L_s : Anten Saçılma Kayıpları (dB)
 L_{sc} : Hedef Saçılma Kaybı (dB)'dir.
 L_{t1} : Havadan Malzemeye Transmisyon Kaybı (dB)
 L_{t2} : Malzemeden Havaya Geri Transmisyon Kayıpları (dB)
 R: Menzil
 V_{peak} : Giriş Üst Değer
 $V_{peak,out}$: Çıkış Üst Değer
 Y: Çıkış
 q_v : Bir noktada var olan elektrik yükü
 μ_r : Relatif manyetik permeabilite
 μ_0 : Manyetik permeabilite
 X: Girdi,
 W: Ağırlık,
 θ : Birleştirme Fonksiyonu,
 $\Delta x, \Delta y, \Delta z$: Aralıklı Izgara Noktaları

Kısaltmalar

ARR: Auto Reload Register (Otomatik Yeniden Yükleme Kaydı)
 ATD: Asenkron Tam Dalga
 CNN: Convolutional Neural Network
 CPU: Central Process Unit
 DYNR: Darbeli Yere Nüfuz Eden Radar
 FASDR: Frekans Adımlı (Modüleli) Sürekli Dalga Radarı
 FDTD: Finite-Difference Time-Domain
 FFT: Hızlı Fourier Dönüşümü
 FPA: Kayan Nokta Birimi
 GUI: Grafiksel Kullanıcı Arayüzü
 GPR: Ground Penetrating Radar

HOG: Histogram of Oriented Gradients
HPC: Yüksek Performanslı Hesaplama
İBYNR: İleri Bakan Yere Nüfuz Eden Radar
KNN: K-En Yakın Komşu
PM: Darbe Modülasyonu
RNN: Recurrent Neural Network
VTK: Visualization Toolkit
TF: TensorFlow
YNR: Yere Nüfuz Eden Radar
YSA: Yapay Sinir Ağları



1. GİRİŞ

Nesne algılama ve tanıma, bir nesneyi görmek ve onunla ilgili tahminlerde bulunmak, farklı özelliklerini seçmek ve sonuçta nesneyi anlamlandırmak için gereken süreçlerdir. İnsanlar bu süreci genellikle üç aşamalı bir yaklaşımla gerçekleştirirler: algılama, tanıma ve tahmin. Ancak, bir makinenin bu sürecin hangi kısmını yapabileceği ya da insanların ulaşamayacağı bir başarımla tüm süreci mükemmel bir şekilde yapabileceği konusunda hala belirsizlikler vardır. Öyle ki yer altı nesnelerini tespit etmeyi istemek daha da güçleşmektedir. Bu alanda birden çok metodlar ve yöntemler vardır. Günümüzde yer altı nesnelerinin tespiti için en popüler çözüm yere nüfuz eden radarları (YNR) kullanmaktır.

YNR sistemleri zemin altı özelliklerinin yüksek çözünürlüklü görüntülenmesi için kullanılan jeofiziksel bir yöntemdir. Toprak altı radarı ya da İngilizce olarak Ground Penetrating Radar (GPR) olarak da bilinir. Elektromanyetik enerjinin patlatılması ve hedeflerden gelen yansımaların alınması süreci sürekli olarak tekrarlanmadan önce bir salvo enerji verir. Hava yönlendirilmiş radarla benzer şekilde çalışır, ancak geniş bir frekans aralığında enerjiyi anında ileterek, çok kısa bir iletim darbesi ve geniş bant bir anten tasarımı kullanarak önemli ölçüde farklılık gösterir (Griffin ve Pippett, 2002). Bu yöntem ile yer altından iki boyutlu ya da üç boyutlu görüntüler alarak çeşitli veriler elde etmek mümkündür. Elde edilecek verilerin işlenmesi verinin çeşidine ve verinin hangi platformlarda işleneceğine göre değişiklik göstermektedir. İşlenen veriler ile saha çalışmalarında gerçek zamanlı tespitler yapabilmek için gömülü sistem üzerinde çıktı almak oldukça önemli bir hale gelmiştir.

Gömülü sistemlerin tasarımı ve optimizasyonu, günümüzde birçok endüstriyel ve akademik alanda yoğun bir şekilde çalışılmaktadır. Bu sistemlerin gömülü olması, genellikle sınırlı kaynaklarla çalışmaları anlamına gelir. Bu sınırlı kaynaklar arasında işlemci gücü, bellek ve veri depolama kapasitesi yer alır. Gömülü sistemlerdeki bu kısıtlamalar, genellikle karmaşık görevleri yerine getirmelerini zorlaştırır ve bazı uygulamalarda daha akıllı algoritmalar gerektirir.

Gömülü sistemlerde yer alan yere nüfuz eden radar (YNR) cihazları, nesne tespiti gibi birçok uygulamada kullanılmaktadır. Ancak, YNR verilerinin elde edilmesi ve analizi karmaşık olabilir ve bu işlem, sınırlı kaynakları olan gömülü sistemlerde gerçekleştirilebilirliği sınırlandırabilir. Bu nedenle, bu tez çalışması, makine öğrenmesi

tekniklerinin YNR verileriyle nesne tespiti yapmak için kullanılabilirliğini araştırmayı amaçlamaktadır.

Bu çalışma için üç farklı nesne ile senaryolar planlanmıştır. Bu nesneler sırasıyla dikdörtgen, küre ve küp olacak şekilde tasarlanmıştır. Bu tasarımlar gprMax-Designer programı ile yapılmış olup; tasarıma ait veriler gprMax programı ile elde edilmiştir.

Bu tez çalışması, yere nüfuz eden radar verileriyle nesne tespiti yapmak için makine öğrenmesi tekniklerinin kullanılabilirliğini araştırmaktadır. Öncelikle, YNR verilerinin özellikleri tanımlanacak ve YNR verilerinin işlenmesi için kullanılan algoritmaların bir analizi yapılacaktır. Daha sonra, makine öğrenmesi teknikleri kullanılarak, nesne tespiti için bir yöntem geliştirilecektir. Bu amaçla, klasik makine öğrenmesi algoritmaları ile derin öğrenme yöntemleri karşılaştırılacaktır. Çalışma, YNR verilerinden nesne tespiti yapmak için makine öğrenmesi tekniklerinin nasıl kullanılabileceğini araştıran öncü bir çalışma olacaktır. Bu çalışma, gömülü sistemler için optimize edilmiş algoritmaların ve makine öğrenmesi tekniklerinin YNR verilerinin işlenmesinde kullanılabilirliğini gösterecektir. Bu çalışma, ileri seviyede bir makine öğrenmesi yöntemlerinin, sınırlı kaynakları olan gömülü sistemlerde de başarılı bir şekilde kullanılabileceğini ortaya koyacaktır.

2. KAYNAK ARAŞTIRMASI

Son yıllarda, araştırmacılar tarafından büyük bir ilgiyle karşılanan bir alan, nesne tespiti konusuyla YNR (Yere Nüfus Eden Radar) verilerinin yapay zekâ kullanımıyla birleşmesidir. Bu gelişen teknoloji, yüksek çözünürlüklü görüntüleme ve son derece hassas nesne tespiti yeteneği sunmuştur. Bu alanda öne çıkan bir çalışmada, Yüksel ve ekibi (2021), YNR verileriyle gerçek zamanlı nesne tespiti yapmanın derin öğrenme yöntemleriyle nasıl gerçekleştirilebileceğini incelemişlerdir. YNR verileri, bu platform üzerinde işlem gören bir derin sinir ağı kullanılarak analiz edilmiş ve tespit edilen nesneler anında görüntülenmiştir. Kullanılan sinir ağı, Convolutional Neural Network (CNN) olarak adlandırılan bir derin öğrenme modelidir. Model, eğitim verileriyle öğrenilir ve daha sonra gerçek zamanlı nesne tespiti için uygulanır. Araştırmanın sonuçları, YNR verileriyle yapay zekâ kullanarak gerçek zamanlı nesne tespitinin başarılı bir şekilde gerçekleştirilebileceğini ortaya koymuştur. Yapılan testler, modelin yüksek doğruluk ve hassasiyet seviyelerine sahip olduğunu açıkça göstermiştir. (Yüksel ve ark., 2021).

Askeri personel veya sivil bireylerin ve askeri/kamu tesislerinin güvenliğini sağlamak ve korumak amacıyla yeraltına gömülü şekilde bulunan El Yapımı Patlayıcıların (EYP) tespiti büyük bir öneme sahiptir. Bu araştırma, öncelikle YNR (Yere Nüfuz Eden Radar) sistemi için geniş bantlı bir Vivaldi Anten (VA) tasarımı içerir. Bu anten tasarımı üretilir ve sistem performansını etkileyen faktörler olan kazanç, giriş yansıma katsayısı ve ışıma örüntüleri detaylı bir şekilde incelenir. Gerçeklenen bu antende 0.4-4 GHz aralığında -10dB' den daha düşük bir geri dönüş kaybı, 2' den daha düşük gerilim duran dalga oranı ve yaklaşık olarak 1.9-5.63 dBi aralığında bir kazanç elde edilmiştir. Anten boyutunu küçültmek ve kazancı artırmak için altıgen açıklıklar ve lineer yarıklar eklenerek Modifiye Vivaldi Anteni (MVA) geliştirilmiş ve üretilmiştir. Bu anten daha sonra Frekans Modülasyonlu Sürekli Dalga (FMCW) radar sistemiyle entegre edilerek YNR sistemi oluşturulmuştur. Bu sistemin yardımıyla, çakıl zeminde 1 cm, 15 cm, 30 cm ve 45 cm derinliklerde gömülü olan 6 farklı EYP, EYP tetikleyici kablolar ve metal mayınlarının tespiti gerçekleştirilmiştir. Ayrıca, EYP dışı gömülü cisim tespiti için 15 cm derinlikte kaya parçası ile deneyler yapılmıştır. Gömülü hedeflerin tespiti, hem VA hem de MVA kullanılarak ayrı ayrı gerçekleştirilmiştir. (Yenikaya ve ark., 2023).

Makine öğreniminin ilk günlerinden bugüne kadar olan süreçte, yapay zekaya olan ilgi sürekli olarak artmıştır. Bu artan ilgi, günümüzde en popüler yapay zekâ algoritmaları olan derin öğrenme mimarilerinin geliştirilmesine yol açmıştır. Derin öğrenme mimarileri ile birlikte, çeşitli yapay zekâ problemlerini çözmek için pek çok yaklaşım geliştirilmiştir. Endüstri, tıp, robotik, görüntü işleme, bilgisayar görüşü, nesne tespiti, ses işleme ve tanıma, çeviri, gelecek tahmini, finans gibi birçok alanda akıllı çözümler ortaya çıkmıştır. Doğan ve arkadaşları bu çalışmada, derin öğrenme mimarilerini ve algoritmalarını inceleyerek, literatürde yapılan araştırmaları göz önünde bulundurarak, farklı uygulama alanlarında elde edilen başarıları değerlendirmiştir. Derin öğrenme mimarilerinin yanı sıra, kullanılan kütüphanelere de değinilmiştir. Ayrıca, çeşitli problem tiplerine yönelik olarak geliştirilen derin öğrenme yaklaşımları da bu çalışmada ele alınmıştır. (Doğan ve ark.,2018)

Nesne tanıma, görüntü işleme, tahmin etme gibi birçok konuya ev sahipliği yapan derin öğrenme gün geçtikçe insanoğlunun ihtiyacı haline gelmeye başlamıştır. Yılmaz ve arkadaşları bu çalışmada derin öğrenme teknikleri kullanılarak nesne tanıma işlemi yaptırılmaktadır. Faster R-CNN (Faster Region Based Convolutional Networks) ağı kullanılarak geliştirilen ve çalışma içerisinde 502 adet görüntü bulunan “Bardak” veri seti oluşturulmuştur. Object Detection API kullanılarak gerçekleştirilen bu çalışmada, farklı epok değerleri ile modeli eğitip, en doğru oranla tahmin yapan epok değeri bulunmaya çalışılmıştır. Gerçekleştirilen 18 ayrı deney üzerinde oluşturulan veri seti üzerinde derin öğrenme ve Faster R-CNN kullanılarak gerçekleştirilmiştir. Eğitim sürecinde en başarılı tahmin oranının bulunması için ise farklı epok sayılarıyla deneyler gerçekleştirilmiştir. (Yılmaz ve ark.,2020)

Kara mayını tespiti için hızlı ve doğru bir yöntemin olması hayati önem taşımaktadır. Online tespitlerde daha da zor hale gelir. Daha iyi CPU performansı için hızlı ve düşük güç tüketen bir algoritma geliştirilmelidir. Bu makale, GprMax programından sentetik veriler kullanarak yeraltı nesnelerini yerleştirip ayırt etmek için üç adımlı bir yöntem önermektedir: n satırlık ortalama kullanarak ön işleme, görüntü ölçeklendirme ve İlgi Bölgesini bir özellik vektörüne dönüştürme. Önerilen yöntem, 7 yöntem, 2 sınıflandırma algoritması ve 3 farklı görüntü ölçeği kullanılarak test edilmiştir. Önerilen yöntem, $K = 5$ için K-En Yakın Komşu (KNN) için Genel Performansı %80.4'ten %90.3'e yükseltmiştir ve Histogramlar Yönlü Gradyanlar % 91.8'e sahiptir. Bununla birlikte, KNN kullanıldığında önerilen yöntemin tespit performansı Histogram of oriented gradients (Yönlendirilmiş Gradyanların Histogramı,

HOG)'a kıyasla biraz daha düşüktür, ancak önerilen yöntemin çalışma süresi performansı ve küçük temsil kapasitesi ile iyi bir potansiyele sahiptir (Meşecan ve ark., 2020).

Son yıllarda elektromanyetik sayısal modellerle çalışmalar yapılmaktadır. Bu çalışmalarda, birçok modelleme tekniği kullanılmakta olup, bu tekniklerden biri olan Finite-Difference Time-Domain (Sonlu-Fark Zaman-Alanı, FDTD) yöntemi oldukça popülerdir. FDTD yöntemi, hesaplama elektromanyetiği alanında yaygın olarak kullanılan bir sayısal modellenme tekniğidir. FDTD yöntemi genellikle, özellikle YNR gibi uygulamalarda, işlem zamanı ve bellek açısından oldukça maliyetli olmaktadır. Bu uygulamalarda, YNR vericisi, alt yapı/ yapı ve hedeflerin hepsi modele dahil edilmeli ve yeterli bir şekilde ayrıştırılmalıdır. Bunun yanı sıra, YNR'nin ileri doğru çözümlenmesi yüzlerce farklı geometri modelinin (A-scan) çözümlenmesini gerektirir. Bu durum, ters YNR probleminin çözümlenmesinde veya makine öğrenme algoritmalarını eğitirken ileri modellerin kullanılması gerektiğinde daha da artar. Warren ve arkadaşlarının yapmış olduğu bir çalışmada, YNR modelleme amacıyla özel olarak tasarlanmış açık kaynaklı bir GPU hızlandırılmış FDTD çözücüsü geliştirilmiştir. NVIDIA'nın CUDA çerçevesini kullanarak GPU için optimize edilmiş kernel tasarımı yapılmıştır. GPU çözücüsü, NVIDIA Kepler ve Pascal mimarilerinde sırasıyla 1194 Mcells/s ve 3405 Mcells/s performans elde etmiştir. Bu, kullanılan bir masaüstü bilgisayarın (Intel Core i7-4790K) paralelleştirilmiş (OpenMP) CPU çözücüsünün elde edebileceği performansın yaklaşık 30 katıdır. NVIDIA GeForce serisi Pascal tabanlı GPU'ların maliyet-performans faydaları, özellikle birçok kişinin yaygın olarak kullanabileceği ortak çalışma istasyonları sayesinde önemli olabilir. Ayrıca, yüksek bant genişliği bellek kullanımı nedeniyle Tesla serisi P100 GPU'nun önemli performans avantajları sunduğunu da belirtmek gerekir. GPU-hızlandırılmalı çözücü performans avantajları, gerçekçi bir GPR senaryosunun büyük ölçekli, (dağılmış ortamlar, engebeli yüzey topografyası ve detaylı anten modeli dahil) simülasyonu ile gösterilmiştir. Sonuç olarak, bu çalışma, GPU tabanlı bir FDTD çözücüsü geliştirerek GPR modellemesi yapmak için daha hızlı ve verimli bir yöntem sunmaktadır. NVIDIA'nın CUDA çerçevesi kullanılarak en uygun çekirdekleri tasarlayarak GPU işlemcileri için optimize edilmiştir. Kepler ve Pascal mimarilerinde sırasıyla 1194 Mcells/s ve 3405 Mcells/s performans oranları elde edilmiştir. Bu, yaygın olarak kullanılan bir masaüstü CPU'sunda (Intel Core i7-4790K) paralelleştirilmiş (OpenMP) CPU çözücüsünün elde edebileceği performans oranlarından 30 kat daha hızlıdır. Bu çalışma ayrıca, oyunculara

yönelik olan NVIDIA GeForce serisi Pascal tabanlı GPU'ların maliyet-performans avantajlarının özellikle dikkate değer olduğunu belirtmektedir. Yüksek bant genişliği bellek kullanımı nedeniyle eşdeğer Tesla serisi P100 GPU ise genel performans avantajları göstermektedir. GPU tabanlı FDTD çözücüsü GPR ortamında gerçekleştirilen bir anti-personel mayın senaryosunun büyük ölçekli ve gerçekçi bir simülasyonunda performans avantajlarını göstermiştir. Bu çalışma, GPR modellemesi için verimli bir yöntem sunmaktadır ve daha ileri araştırmaların yapılmasına olanak tanır (Warren ve ark., 2019).

Literatürde GPR ile gömülü nesne tespiti üzerine yapılan çalışmalara bakıldığında GPR sistemlerinde birçok farklı teknik kullanılmıştır. Bu tekniklerin her biri farklı koşullarda avantajlar sunmaktadır. Razali ve arkadaşları yapmış oldukları bir çalışmada düşük frekanslarda çalışan bir GPR sistemini kullanarak yer altı gömülü nesnelerin yüksek frekanstan daha derinde yer yüzeyinden tespit edilmesini sağlamışlardır. Bununla birlikte, darbe modülasyonu (PM) tekniği ve yüksek frekans taşıyıcısı kullanılarak sistemden üretilen çıkış sinyalinin, çok sayıda yüksek dalgalanma sinyaline sahip olduğunu görmüş ve sonuç olarak bulanık bir görüntüye neden olduğunu tespit etmişlerdir. Bununla birlikte, bir zarf detektörü yöntemiyle sinyali yeniden işleyerek bu dalgalanmayı en aza indirmişlerdir. Yapmış oldukları çalışmada, GPR sisteminde ArJED© asenkron tam dalga (ATD) adı verilen bir algılama tekniği kullanmış ve 0,06 ila 0,08 GHz frekans aralığında test etmişlerdir. YNR sisteminin gömülü nesne algılama sensörü olarak bir dipol anten kullanmışlardır. Çalışmış oldukları algılama sistemi, 2 cm, 5 cm, 7 cm ve 20 cm ile başlayan dört derinliği içerir. Gömülü nesne görüntülerinin zarf algılama tekniğinin uygulanmasından önceki ve sonraki karşılaştırması yapılmış ve önermiş oldukları zarf algılama tekniğinin YNR sisteminin daha net bir radargram görüntüsünü ürettiğini kanıtlamışlardır (Razali ve ark., 2022).

Gözde Altın tarafından yapılan bir çalışmada, GPR B Tarama görüntülerini etkin, hızlı ve tam otomatik olarak bölütlenebilen bir görüntü işleme algoritması önerilmiştir. Bu algoritma, segmentasyon ve Hough dönüşümü temel alınarak tasarlanmıştır. Algoritmanın temel amacı, GPR B tarama görüntülerinde hiperbol olarak görünen gömülü cisimlerin daha belirgin hale getirilebilmesi için görüntülerdeki gürültüyü azaltmaktır. Bu işlem denetimsiz olarak gerçekleştirilirken, toprak düzlemi ile gömülü cisim hiperbollerini birbirinden ayırmak gerekmektedir. Önerilen algoritma, 15 adet simülasyon ve 11 adet gerçek GPR B Tarama görüntüleri üzerinde test edilmiştir

ve gösterdiği yüksek performans sayesinde birçok GPR B Tarama görüntüleri içinde kullanılabileceği sonucuna varılmıştır. Algoritmanın performans zamanı, literatürdeki diğer çalışmalarla direkt olarak kıyaslanamasa da kendi içerisinde yüksek performans göstermiştir. Bu nedenle, önerilen görüntü işleme algoritmasının gerçek zamanlı sistemlere uygulanabileceği belirtilmektedir. Ayrıca, daha kapsamlı çalışmalar yapmak için farklı senaryolar içeren ve daha fazla sayıda GPR B Taraması görüntüleri kullanarak derin öğrenme ve denetimli görüntü işleme algoritmaları kullanılarak daha gelişmiş algoritmalar geliştirilebilir. Bu algoritma kullanılarak, GPR B Tarama görüntüleri üzerinden gömülü nesne şekil ve malzeme tespiti yapılabileceği sonucuna varılmıştır.



3. YERE NÜFUZ EDEN RADAR

Yere nüfuz eden radar (YNR) sistemleri, son zamanlarda arkeoloji, jeoloji, inşaat mühendisliği ve savunma alanlarında elektromanyetik dalgaları kullanarak toprak, beton, tuğla, ağaç gibi engellerin arkasına gizlenmiş hedeflerin uzaktan tespitini sağlayabilen bir radar çeşididir (Çolak, 2018). Mühendislik uygulamaları arasında gömülü yapıların, tünellerin, çöplüklerin ve kirletici bulutların bulunması ve test edilmesi yer alır. YNR ayrıca arkeolojide ve mayın tespitinde de kullanılır (Saleh, 2011).

3.1. Yere Nüfuz Eden Radarlara Genel Bakış

Elektromanyetik sinyallerin uzaktaki karasal metal nesnelerin varlığını belirlemek için ilk kullanımı genellikle 1904'te Hiilsmeyer'e atfedilir. Ancak gömülü nesnelerin konumu için kullanımlarının ilk açıklaması altı yıl sonra Leimbach ve Lowy'nin bir Alman patentinde ortaya çıktı. Teknikleri, dipol antenlerini bir dizi dikey sondaj deliğine gömmekten ve ardışık çiftler iletmek ve almak için kullanıldığında alınan sinyallerin büyüklüğünü karşılaştırmaktan oluşuyordu. Bu şekilde, çevreleyen ortama göre daha yüksek iletkenliği sayesinde tercihen radyasyonu emen dizi içindeki herhangi bir bölgenin kaba bir görüntüsünü oluşturmaktaydı. Yapılan çalışmaların ana özellikleri, yeraltı özelliklerinden dolayı perdeleme veya kırınım etkilerinin kullanılması ve saçılma üretmek için iletkenlik varyasyonlarına güvenmekti. Bu özellikler madenler için tamamen gerçekleşmeyecek uygulamalara yönelik olanlar da dahil olmak üzere bir dizi başka patent açıklamalarında da mevcuttu (Daniels, 2004). 1926'da Hiilsenbeck'in çalışması, gömülü özelliklerin yapısını belirlemek için darbeli tekniklerin ilk kullanımı gibi görünmektedir. Yapmış olduğu çalışmada, iletkenliği içermesi gerekmeyen herhangi bir dielektrik varyasyonun da yansımalar üreteceğini ve yönlü kaynakların daha kolay gerçekleştirilmesi yoluyla tekniğin sismik yöntemlere göre avantajları olduğunu kaydetti (Anderson ve ark., 2001).

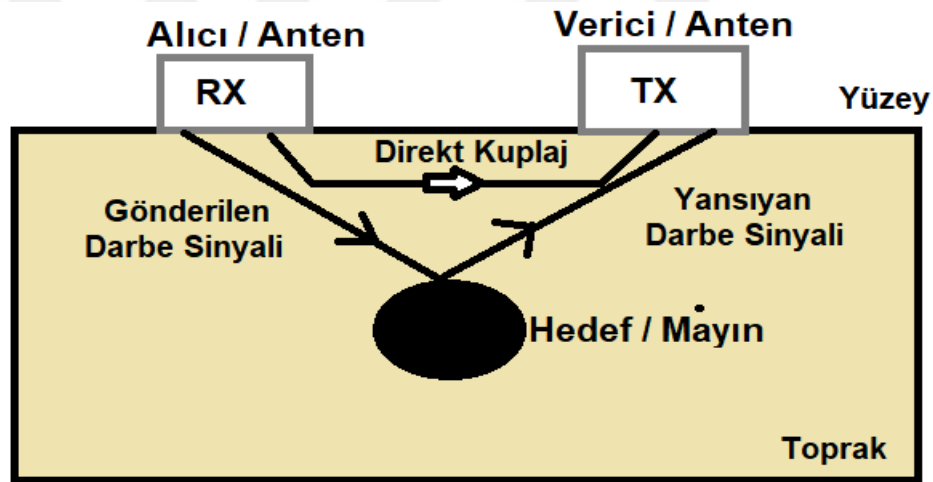
3.2. Yere Nüfuz Eden Radarların Çalışma Prensipleri

YNR, gezegen araştırmalarından gömülü madenlerin tespitine kadar çok geniş bir uygulama alanına sahiptir. Bir dizi frekans işleminin seçimi, belirli bir modülasyon şeması, anten tipi ve polarizasyonu, hedefin boyutu ve şekli, araya giren ortamın iletim

özellikleri ve operasyonel gereksinimleri dahil olmak üzere bir dizi faktöre bağlıdır (Lo ve Inderwiesen, 1994). Yapılabilecek çalışmalar belirli bir sistem tipinin özellikleri, algılama ve çözünürlüğü etkileyen çeşitli faktörlerin incelenmesiyle hazırlanabilir. Sistemin başarılı bir şekilde çalışması için, yere nüfuz eden radarın aşağıdaki unsurları sağlaması gerekmektedir:

- 1) yeterli bir sinyal / dağınıklık oranı
- 2) yeterli bir sinyal / gürültü oranı
- 3) hedefin yeterli bir uzaysal çözünürlüğü
- 4) hedefin yeterli bir derinlik çözünürlüğü.

Kuyu tomografik radar görüntülemeye ileri iletim yöntemleri kullanılmasına rağmen, çoğu YNR sistemi hedeften geri saçılan sinyali algılar. Şekil 1.1’de bu çalışma mantığı gösterilmiştir.



Şekil 1.1. YNR Sistemi Basit Çalışma Gösterimi

Yere nüfuz eden radarlar için temel elektromanyetik prensipleri anlamak ve açıklamak oldukça önemlidir. Bu prensipler, radar teknolojisinin temelini oluşturur ve yere nüfuz eden radar sistemlerinin çalışmasının anlaşılmasını sağlamaktadır. Bu prensipleri anlamak, radar sistemlerinin etkili ve doğru çalışmasını sağlamak için gereklidir.

Faraday Kanunu (Faraday's Law)

$$\oint_C \mathbf{E} \cdot d\mathbf{l} = - \iint \frac{\partial \mathbf{B}}{\partial t} \cdot d\mathbf{s} \quad (3.1)$$

Manyetik akı, yüzeyin sınırlı integrali ile ilgilidir ve elektromotor kuvvet yaratır.

Bir mıknatıs boyunca bobini hareket ettirdiğimizde ve elektromotor kuvveti yarattığımızda bu formül kullanılır.

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad (3.2)$$

Bir kapalı devredeki manyetik akı yoğunluğunda meydana gelen değişim, bu devre içinde bir elektrik alanın dönmesine neden olur. Yani, manyetik alanın değişimi elektrik alanın oluşmasına neden olur. Bu prensip elektromanyetik indüksiyon olarak adlandırılır ve elektrik jeneratörlerinin çalışma prensibini açıklar.

Ampere-Maxwell Kanunu (Ampere-Maxwell Law)

$$\oint_C H \cdot dl = \iint \frac{\partial D}{\partial t} ds + \iint J_c \cdot ds + \iint J_s \cdot ds \quad (3.3)$$

Eğer iletken üzerinden geçen bir akım varsa, manyetik alanını ve dönme yönü elde edilir (sağ el kuralı).

$$\nabla \times H = \frac{\partial D}{\partial t} + J_c + J_s \quad (3.4)$$

Bu denklem elektromanyetizmanın temel yasalarından biridir ve manyetik alanın (H) dönmesini manyetik akı yoğunluğunun (D) zaman türevi, serbest yüzey akım yoğunluğu (J_c) ve iletkenlik akım yoğunluğu (J_s) ile ilişkilendirir.

Gauss Elektrik Kanunu (Gauss' Electric Law)

$$\oiint_S D \cdot ds = \iiint q dV \quad (3.5)$$

Kapalı bir yüzey hacminde elektrik akısı vardır. Hiçbir yükün olmadığı ve hiçbir kaynağın olmadığı yerlerde sıfır olur.

$$\nabla \cdot D = q_v \quad (3.6)$$

Elektrik akısının yüzdesi bir noktadaki yüke eşittir veya sıfırdır, eğer yüküm yoksa manyetik yoğunluk sıfırdır.

Elektrik Yükünün Sürekliliği (Continuity of Electric Charge)

$$\oint_S J \cdot ds = \iiint \frac{\partial q}{\partial t} dV \quad (3.7)$$

Akımın kaybedilmeyeceğini veya yaratılamayacağını söyleyen şey, elektrik haritalarının sürekliliğini veren bu denklemdir.

Stoke ve Diverjans Teoremleri (Stoke's and Divergence Theorems)

$$\oint_S A \cdot dl = \iint (\nabla \times A) \cdot ds$$

$$\oint_S A \cdot ds = \iiint (\nabla \cdot A) dV \quad (3.8)$$

Bir vektör miktarının kapalı integralini yüzey boyunca rotasyoneliyle ve kapalı yüzey yüzey etrafındaki integral aracılığıyla ilişkilendirilir.

Alanlar İçin Durumsal İlişkiler (Constitutive Relations for fields)

$$D = \epsilon \cdot E \quad (3.9)$$

:Elektriksel alandaki (E) elektrik akı yoğunluğunu (D) ile ilişkilendiren denklemdir. ϵ birincil dielektrik sabit, E elektriksel alanı temsil eder.

$$B = \mu \cdot H \quad (3.10)$$

Manyetik alandaki (H) manyetik akı yoğunluğunu (B) ile ilişkilendiren denklemdir. μ manyetik permeabilite, H manyetik alanı temsil eder.

$$J_c = \sigma \cdot E \quad (3.11)$$

İletken malzemelerdeki akım yoğunluğu (J_c) elektriksel alana (E) orantılıdır. σ iletkenlik katsayısıdır.

$$\epsilon_r = \frac{\epsilon}{\epsilon_0} \quad \mu_r = \frac{\mu}{\mu_0} \quad (3.12)$$

Relatif dielektrik sabit (ϵ_r) ve relatif manyetik permeabilite (μ_r) terimleri, boşluk veya serbest uzaydaki dielektrik sabiti (ϵ_0) ve manyetik permeabilite (μ_0) ile orantılıdır.

Sınır Koşulları (Boundary Conditions)

$$\nabla \times H = \frac{\partial D}{\partial t} + J_c + J_s \quad (3.13)$$

Manyetik alanın (H) dönmesi, elektrik akı yoğunluğunun (D) zaman türevi ve akım yoğunluğu terimleri ile ilişkilidir.

$$I_y = \frac{I_s(t)\Delta y}{\Delta x\Delta z} \quad (3.14)$$

Bu denklem, bir iletken telin taşıdığı akımın ($I_s(t)$) y eksenini boyunca yoğunluğunu ifade eder. Δx ve Δz , x ve z eksenini boyunca alan yoğunluğunu temsil eder.

Bu denklemler, elektromanyetik olayların temel kurallarını ve alanların davranışını anlamada kullanılır. Elektromanyetizma alanında önemli bilgiler sunar ve birçok farklı uygulama alanında kullanılır.

3.2.1. Yere Nüfuz Eden Radar Sistemlerinde Karşılaşılan Kayıplar

Radarın menzile performansının ilk tahmini, bazı faktörleri dikkate alarak elde edilebilir. Bu faktörler; yoldaki kayıplar, hedef yansıması, dağınıklık ve sistemin dinamik menzile mesafesidir. Radarın uzaysal çözünürlüğü, derinlik ve plan çözünürlüğü ayrı ayrı dikkate alınarak belirlenebilir. YNR sistemlerinin çoğu bir darbe zaman alanı dalga biçimi kullanır ve yansıyan sinyali bir örnekleme alıcısında alır. Bununla birlikte, son yıllarda Frekans Modüleli (Adımlı) Sürekli Dalga Radarı (FASDR) ve kademeli frekans radar modülasyon şemalarından daha fazla yararlanılmıştır (Daniels, 2004). Bileşenlerin maliyeti azaldıkça, dinamik aralıkları olacak şekilde tasarlanabileceğinden bu sistemlerin daha fazla kullanılması beklenmektedir.

3.2.1.1. Yol Kayıpları

Yukarıda bahsedildiği üzere YNR menzile performansı yol kaybı, hedef yansıması, dağınıklık ve sistemin dinamik menzile mesafesine bağlıdır. Yol kaybı belirli mesafeler için aşağıdaki eşitlik ile verilir.

$$LT = L_e + L_m + L_{t1} + L_{t2} + L_s + L_a + L_{sc} \quad (3.15)$$

Burada;

L_e = anten etkinlik kayıpları (dB)

L_m = anten uyum kayıpları (dB)

L_{t1} = havadan malzemeye transmisyon kaybı (dB)

L_{t2} = malzemeden havaya geri transmisyon kayıpları (dB)

L_s = anten saçılma kayıpları (dB)

L_a = malzemenin zayıflama kaybı (dB)

L_{sc} = hedef saçılma kaybı (dB)'dir.

3.2.1.2. Hedef Yansıması

Hedefin saçılma karakteristiğinden bağımsız olarak YNR görüntüsünü etkileyen sinyallere hedef yansıması (clutter) denir. Klasik radarlarda olan kargaşalardan farklıdır. Klasik radarlarda, hareketli hedef göstergesi ile filtrelenir. Ancak bu çözüm YNR'ye uygulanabilir değildir. Antenden gelen sinyalin yer yüzeyindeki kırılmaları çevre yankısına sebep olabilir. Çevre yankısı da anten konfigürasyonuna göre değişir. Kırılma seviyesinin daha sabit olduğu paralel düzlemsel dipol anten çevre yankısına karşı tercih edilir (Daniles, 2004).

3.2.1.3. Dağınıklık

Toprak ve zemin yansıma kaynaklarının karakteristik empedansındaki kısmi değişiklikler dağınıklığa neden olabilir. Öte yandan, antenin yan kulakçığında, çoğu zaman yüzeyin üzerindeki hedeflerden yansımalar sorunlara neden olabilir. Bu problem, antenin yan ve arka hüzmeye ışınmasını zayıflatarak aşılabilir (Şengül 2018).

3.3. Yere Nüfuz Eden Radar Sistemlerinde Kullanılan Teknikler

3.3.1. Frekans Adımlı Sürekli Dalga Radar

Frekans Adımlı Sürekli Dalga Radar (FASDR), yeraltına belirli bir frekans aralığında elektromanyetik dalgalar yayar. Bu sayede yansıyan sinyalin faz farkını ve hedefin yansıma katsayısını ölçerek görüntüleme gerçekleştirir. Frekans alanında çalışan sistemlerin yüksek sinyal-gürültü oranlarına ulaşması daha kolay olsa da, darbe dalgası üretiminin süre alanında çalışan sistemlerden daha pahalı ve tasarım açısından daha karmaşıktır (Tuncer, 2011; Orhan, 2016).

FASDR radar sistemi sürekli değişen taşıyıcı dalgayı gerilim kontrollü osilatör üzerinden seçilen frekansla gönderir. Alınan sinyal, gönderilen dalga şekliyle karıştırılır ve alınan sinyalin fazıyla ilişkili farklı bir frekansla sonuçlanır. FASDR'ında verici

zamanın bir fonksiyonu olarak değişir. Eğer değişim doğrusal ise ve hedeften Td zamanında döndüğü kabul edilirse;

$$T_d = \frac{2R}{c} \quad (3.16)$$

olarak elde edilir. Burada, R: menzil, c: ışık hızıdır (Alper, 2011).

3.3.2. Darbeli Yere Nüfuz Eden Radar

Gömülü nesnelerin araştırılmasında, ilk olarak darbeli radarın (pulsed radar) kullanımı Hiilsenbeck tarafından gerçekleştirilmiştir (Anderson ve ark., 2001). Darbeli yere nüfuz eden radar (DYNR) sistemi, yaklaşık genliği olan değişken genişlikte (yaklaşık 1,5ns - 6ns) darbeler üreten bir çalışma sergiler. Darbe sinyalinin şekli bant genişliğini belirler. Hedef, alıcıya gelen sinyalin genliği ve zaman gecikmesi ile tespit edilir

$$V_{\text{peak}} = 20V \quad (V_{\text{peak,out}} = 10\log_{10} \frac{20V^2 / 50\Omega}{1 \text{ mW}} = 39\text{dBm}) \quad (3.17)$$

olacak şekilde, bir verici antenini (Tx) besler. Genellikle birçok çalışmada alıcı kısım, alıcı anten (Rx) vasıtasıyla yakalanan sinyalleri örnekleyen, kayıt ünitesi olarak kullanılan bir dijital osiloskop içerir. Osiloskobun farklı tam ölçekli dikey ayarları için DYNR sisteminin algılama yeteneği hakkında daha genel bilgiler elde etmek ve aynı zamanda elde edilecek örneklemeden önce birden fazla tekrarlanan A-Taraması ve sinyal amplifikasyonu üzerinden ortalama alınır (Seyfried ve Schoebel, 2015).

3.3.3. İleri Bakan Yere Nüfuz Eden Radar

Gömülü ve karaya yerleştirilmiş tanksavar mayınlarının verimli bir şekilde tespit edilmesi için gereken optimal radar parametrelerinin tanımlanmasına yardımcı olmak için yüksek çözünürlüklü ileri bakan yere nüfuz eden radar (İBYNR) gömülü sistemleri tasarlanmıştır. Bu radar ileriye dönük olduğundan, bir araca ihtiyaç duyar ve sentetik açıklık, anten elemanlarının hareket yönüne dik hareket etmesiyle oluşturulur. Çoğu zaman gönderme ve alma antenlerinin her biri için iki adet 1 m, çift doğrusal polarizasyon, dört kenarlı horn kullanılır. Her bir anten çifti, veri toplama sırasında sentetik açıklık dizisini üretmek için anten boynuzları mekanik olarak yana doğru hareket ettiren bilgisayar kontrollü doğrusal bir sürücüye monte edilir. Antenler, radar minibüsünün ön üst kısmına üst üste monte edilir (Kositsky ve Milanfar, 1999). Tespit ve yanlış alarm istatistiklerini iyileştiren algoritmalar kullanarak dağınıklığı bastırma ve

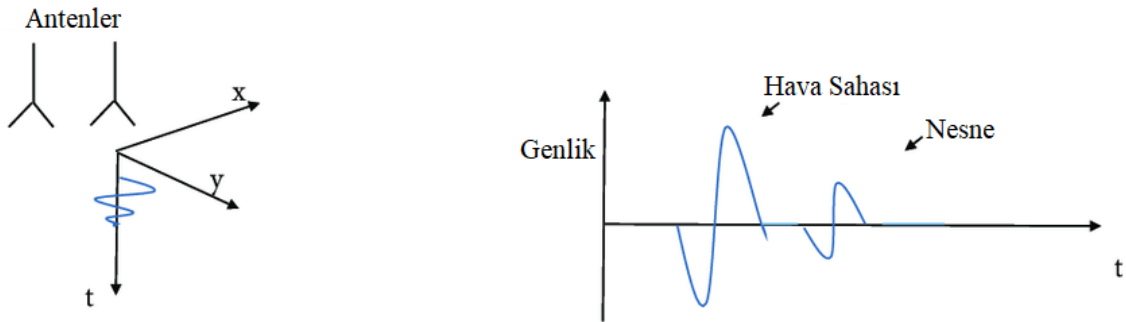
hedef tanıma konusunda diğer radar çeşitlerine göre daha avantajlıdır (Kositsky ve ark., 2002; Ozakin ve ark., 2015).

3.4. Yere Nüfuz Eden Radar Veri Tipleri

YNR verileri farklı formatta toplanıp, görüntülenebilir. Bu veriler A, B ve C-tarama olmak üzere 3 farklı şekilde işlenmekte ve görüntülenebilmektedir. YNR verilerinin gösterimi için kullanılan tarama çeşitleri sistemin ihtiyacına, tarama amacının çeşidine ve tarama alanına göre değişmektedir.

3.4.1. A-Tarama

Antenlerle sabit bir tarama konumunda (x_i, y_j) kaydedilen tek bir veri dalga biçimi $d(x_i, y_j, t)$, bir A-taraması olarak adlandırılır. YNR için bir örnek A-taraması Şekil 3.2'de gösterilmiştir. Tarama pozisyonu sabit bir A-taraması olduğundan, bu sadece zamanın bir fonksiyonudur ve zaman gecikmesi hedefin derinliği ile ilgilidir. A-taramaları ve enerjileri genellikle ilgili tarama pozisyonlarında hedef tespit testleri için kullanılır (Gurbuz ve Duman, 2015).

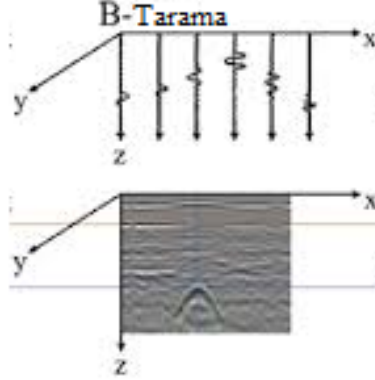


Şekil 3.2. YNR A-tarama verilerinin gösterimi

3.4.2. B-Tarama

YNR B-tarama görüntülerinde, gömülü nesneler elektromanyetik dalgalar aracılığıyla ortaya çıkarılır ve radagram adı verilen iki boyutlu bir grafikte hiperbolik modeller olarak görselleştirilir. YNR için bir örnek B-taraması Şekil 3.3'de gösterilmiştir (Merkle ve ark., 2021). Radagramların yorumlanması genellikle zordur ve toprak altındaki gömülü nesneleri doğru bir şekilde tanımlamak için bu alandaki uzmanların yardımı zorunludur (Ozkaya ve ark., 2020). Son yıllarda, YNR B-tarama

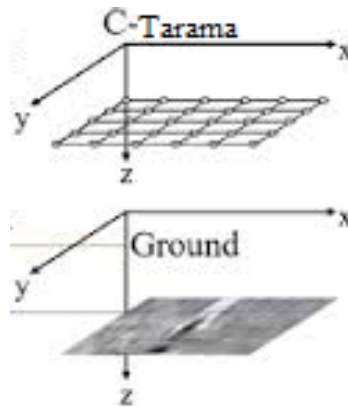
görüntülerinde hiperbolik paternlerin analizi için birçok denetimli ve denetimsiz yöntem kullanılmıştır. Özellikle, YNR B-tarama görüntüleri analizi, gömülü nesnelerin tespiti ve tanımlanması için birçok yaklaşım sunar. (Parillo ve ark., 2005).



Şekil 3.3. YNR B-tarama verilerinin gösterimi

3.4.3. C-Tarama

Pratik olarak, C-tarama görüntüleme modu, belirli bir toplama zamanında bir genlik haritası sağlar. Bir alan, örneğin düzenli bir ızgara boyunca veri toplayarak araştırıldığında, C-tarama görselleştirme modu, sinyal genlikleri açısından yüksek bir yansıtma ile karakterize edilen homojen olmayan noktaların saptanmasında diğer tarama yöntemlerine göre çok yararlı olabilir. YNR için bir örnek C-taraması Şekil 3.4'de gösterilmiştir. C-tarama görselleştirmesi, işlem yalnızca uzayda çalışacak şekilde tanımlanmış bir zamansal an ile ilgilidir. Yolların önemli ölçüde genişlemesi ve bunun sonucunda gerekli olan büyük miktarda veri işleme nedeniyle, C-tarama görselleştirme modu geniş çaplı uygulamalarda nadiren kullanılır (Benedetto ve ark., 2016).

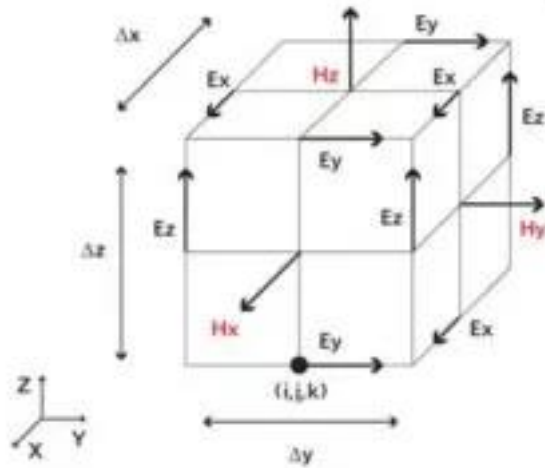


Şekil 3.4. YNR C-tarama verilerinin gösterimi

3.5. Yere Nüfuz Eden Radarın Bilimsel Gösterimi ve Veri Seti

3.5.1. GprMax

GprMax, YNR'nin sayısal modellemesi için elektromanyetik dalga yayılımını simüle eden açık kaynaklı bir yazılımdır ve <http://www.gprmax.com> adresinden temin edilebilir. Bu yazılım YNR simülasyonu için tasarlanmış olup mühendislik, jeofizik, arkeoloji ve tıp gibi alanlarda elektromanyetik dalga yayılımını modellemek için kullanılabilir. Köprüler, yollar gibi kritik altyapıların değerlendirilmesi, gömülü hizmetlerin bulunması, buzulların haritalanması, anti-personel mayınların bulunması ve insan vücudundaki tümörlerin tespiti gibi geniş bir uygulama yelpazesi vardır. GprMax, Python dilinde yazılmış komut satırı tabanlı bir yazılımdır ve performans açısından kritik olan bölümleri Python ile yazılmıştır. Şu anda grafiksel bir kullanıcı arayüzü (GUI) bulunmamaktadır, bu da esnek ve betiklenebilir bir yazılım olmasını sağlar ve yüksek performanslı hesaplama (HPC) ortamlarında, yani süper bilgisayarlarda çalışabilir. GprMax, Maxwell denklemlerini, sonlu farklar zaman alanı (Finite-Difference Time-Domain, FDTD) yöntemini kullanarak 3D'de çözmek için kullanır. FDTD yöntemi, gibi birçok yayında açıklanan bir diferansiyel denklem tabanlı bir çözücüdür.



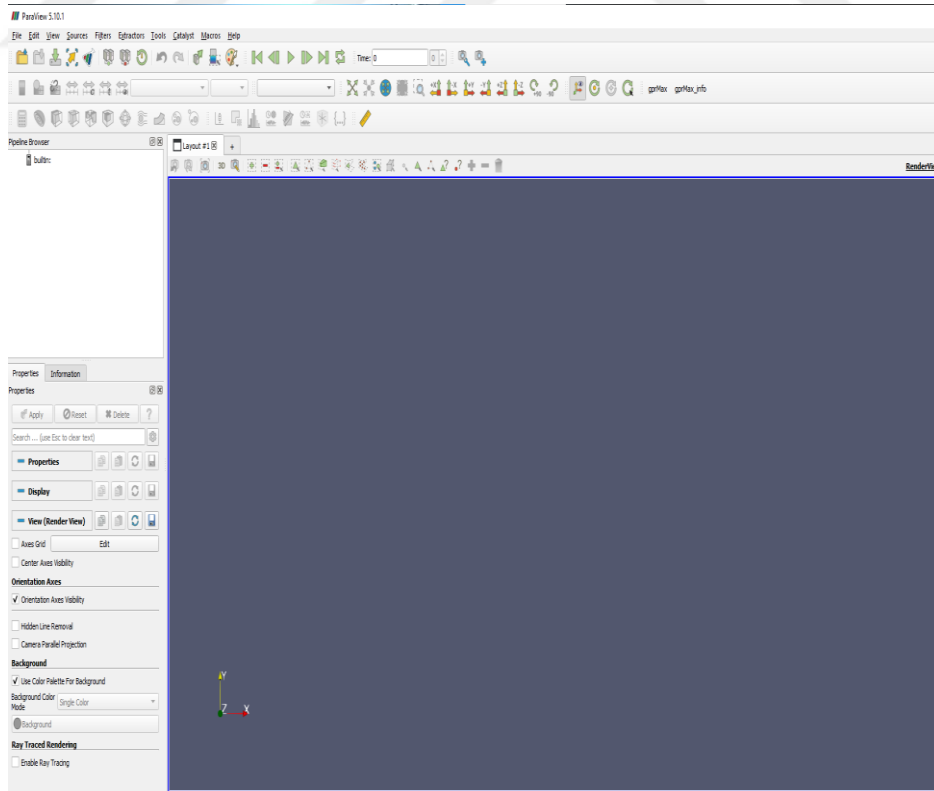
Şekil 3.5. FDTD Hücre Boyutu Δx , Δy , Δz . (Yee, 1966)

FDTD yöntemi, Maxwell denklemlerini bir ağ üzerinde çözer. E ve H'yi Δx , Δy ve Δz aralıklı ızgara noktalarında, E ve H üç uzamsal boyutun hepsinde iç içe geçmiş olarak hesaplar ve saçılma, iletim, yansıma, absorpsiyon vb. etkileri içerir (Şekil 3.5). FDTD, zaman alanlı bir çözümdür, ancak Hızlı Fourier Dönüşümü (FFT) ve Ayırık Fourier Dönüşümü (DFT) kullanılarak frekans analizi de mümkündür (Berenger, 1994).

Özetlemek gerekirse, FDTD yönteminin avantajları basit, tamamen açık, genel ve sağlam bir teknik olmasıdır. Temel zayıflık ise tüm hesaplama alanının ayrıklaştırılması gerektiğinden dolayı kapsamlı hesaplama kaynakları gerektirebilmesidir. FDTD yönteminin zamanda alan doğası, tek bir simülasyonda geniş bir frekans aralığının modellenenebilmesini sağlar. Bu, genellikle ultra geniş bant (UWB) olan YNR sistemlerinin simülasyonu için çok uygundur. Bununla birlikte, hesaplama alanı yine de ilgi duyulan en yüksek frekansa göre ayrıklaştırılmalıdır (Warren ve ark., 2019). GprMax, CPU veya GPU üzerinde çalıştırılabilir. CPU çözücüsü, çok çekirdekli CPU'lar üzerinde çalışmasını sağlayan OpenMP ile paralelleştirilmiştir. GPU çözücüsü, NVIDIA CUDA programlama modeli kullanılarak geliştirilmiştir. GprMax ayrıca, CPU düğümleri veya çoklu GPU'larla çalışabilen bir Mesaj İletişim Arayüzü (Messaging Passing Interface, MPI) görev çiftliği özelliğine sahiptir (Anonim, 2023).

3.5.2. Paraview Programı

Tez çalışmasında bilimsel verilerin görselleştirilmesi bu program üzerinden sağlanmış olup programa ait arayüz aşağıdaki Şekil 3.6'daki gibidir.



Şekil 3.6. Paraview Programının Arayüzü

ParaView, popüler bir açık kaynaklı genel amaçlı bilimsel veri kümelerini görselleştirmek için tasarlanmış bir uygulamadır. ParaView, donanım hızlandırmalı paralel görselleştirme desteği sağlar ve seviye detayı teknikleri aracılığıyla etkileşimli görselleştirme performansı elde eder. ParaView, dağıtık ve paylaşımlı bellekli paralel makinelerin yanı sıra tek işlemcili PC üzerinde çalışır ve Windows, Mac OS X, Linux ve çeşitli Unix istasyonları, kümeleme sistemleri ve süper bilgisayarlar üzerinde başarıyla test edilmiştir. ParaView, veri işleme ve görselleştirme motoru olarak Görselleştirme Araç Kiti (Visualization Toolkit, VTK) kullanır ve kullanıcı arayüzü Qt kullanılarak yazılmıştır. ParaView, katmanlı bir mimari olarak tasarlanmıştır. Temel katman VTK'dır ve veri temsilleri, görselleştirme algoritmaları ve bu temsilleri ve algoritmaları birleştirerek çalışan bir program oluşturmak için bir mekanizma sağlar. İkinci katman, VTK'ya paralel uzantılardır. Bu katman, veri akışı ve paralel hesaplama desteğini sağlamak için VTK'yı genişletmiştir. Bu uzantılar şu anda araç kitinin bir parçasıdır. ParaView, üç katmanlı bir istemci-sunucu mimarisi olarak tasarlanmıştır. Veri Sunucusu, veri okuma, filtreleme ve yazma işlemlerinden sorumlu bir birimdir. Pipeline tarayıcısında görülen tüm pipeline nesneleri veri sunucusunda bulunur. Render Sunucusu, görselleştirmelerden sorumlu bir birimdir. İstemci, görselleştirmenin kurulmasından sorumlu bir birimdir. İstemci, GUI kullanarak nesne oluşturma, yürütme ve yok etme işlemlerini kontrol eder. Bu mantıksal birimlerin fiziksel olarak ayrılmış olması gerekli değildir. En yaygın istemci-sunucu modunda, pvserver programı paralel bir makinede çalıştırılır. Pvserver programında veri sunucusu ve render sunucusu bir araya getirilmiştir. İstemci ve sunucu bir soket üzerinden bağlanır ve iletişim için göreceli olarak yavaş bir mod olduğu varsayıldığından, bu soket üzerinden veri aktarımı minimize edilir (Kačeniauskas ve ark., 2010). ParaView içinde mevcut olan görselleştirme aracından biri, düzensiz örgülerin hacim görselleştirmesidir. Hacim görselleştirme, bir örgüyü yarı saydam bir katı olarak görselleştirerek kullanıcının üç boyutlu uzayda her noktayı aynı anda görmesine olanak tanır. Hacim görselleştirme hesaplama açısından yoğun olduğundan, ParaView programı, süreçleri hızlandırmak için bir paralel görselleştirme algoritması kullanmaktadır. Paralel görselleştirme algoritması çok esnek bir yapıya sahiptir (Peterka ve ark., 2009). Hem hacimler hem de yüzeyler için aynı derecede iyi çalışır ve bir hacim ile opak çokgen yüzeylerin kesişimini doğru bir şekilde görselleştirebilir. Paralel görselleştirme algoritması ayrıca döşemeli ekranlar için görüntüler oluşturabilir. (Moreland ve ark., 2007).

4. GÖMÜLÜ SİSTEMLER

Gömülü sistem, genellikle donanım ve mekanik bileşenleri de içeren daha büyük bir sisteme bileşen olarak gömülmüş olan, belirli bir görev veya amaç için tasarlanmış özel bir bilgisayar sistemidir. Gömülü sistemler terimi, genellikle daha büyük sistemlerin bir parçası olduğu için kullanılır. Bir anlamda, daha büyük sistemin içine yerleştirilmiş küçük bir sistemdir. Gömülü sistemler çoğu durumda tamamen entegredir, yani sistemlerin içinde yer alan sistemlerdir. Genellikle kendi başlarına çalışamazlar. Cep telefonlarından navigasyon araçlarına, video kameralardan arabalara ve ev aletlerine kadar birçok farklı uygulama alanı vardır. Gömülü sistemlerin işlevselliği ve karmaşıklığı önemli ölçüde artmıştır, bu da gömülü sistem ürünlerini verimli bir şekilde geliştirmeyi giderek zorlaştırmaktadır. Benzer bir eğilim yazılım mühendisliğinde de görülmekte olup, geleneksel plana dayalı metodolojilerin bu artan karmaşıklık ve öngörülemezlikle başa çıkamadığı ve buna tepki olarak hafif ve esnek yöntemlerin yazılım ürün geliştirmede geniş popülarite kazandığı görülmektedir (Kaisti ve ark.; Barr, 2019).

Gömülü sistemler, geleneksel yazılım uygulama geliştirmeden birçok farklı şekilde ayrılabilir, gömülü alanda esnek yöntemlere ilişkin giderek artan bir farkındalık vardır. Bununla birlikte, bu bilgi çeşitli forumlarda dağınık bir şekilde bulunmakta ve ayrıca oldukça tutarsız olmaktadır. Gömülü sistemlerin geliştirilmesi, genellikle daha büyük bir sistemin veya cihazın bir parçası olan yazılım ve donanımın geliştirilmesini içerir (Kaisti ve ark., 2013).

4.1. STM32F407VGT6

STM32F407xx ailesine ait mikrodenetleyiciler düşük güç tüketimi ve yüksek performans hedeflemektedir. 168 MHz'e kadar frekansta çalışan yüksek performanslı ARM (Advanced RISC Machines) Cortex-M4 32-bit RISC çekirdeğini temel alır. Cortex-M4 çekirdeği, tüm ARM tek duyarlıklı veri işleme talimatlarını ve veri türlerini destekleyen bir Kayan nokta birimi (FPU) tek duyarlık özelliğine sahiptir. Ayrıca, uygulama güvenliğini artıran eksiksiz bir DSP (digital signal processing) yönergesi seti ve bir bellek koruma birimi (MPU) uygular.



Şekil 4.1. STM32F407VGT6 Mikroişlemcisi

Bu mikrodeneleyici üç adet 12-bit ADC, iki DAC, bir düşük güçlü RTC, motor kontrolü için iki PWM zamanlayıcı, iki genel amaçlı 32-bit zamanlayıcı dahil olmak üzere on iki genel amaçlı 16-bit zamanlayıcı sunar. Ayrıca standart ve gelişmiş iletişim arayüzlerine sahiptirler (STMicroelectronics, 2016). Bu çalışmada STM32F407VGT6 mikroişlemcisi barındıran STM32F407G-DISC1 geliştirme kartı kullanılmıştır (Şekil 4.1). Bu geliştirme kartına ait özellikler aşağıdaki gibi listelenmiştir.

- LQFP100 paketinde bulunan 32-bit Arm® Cortex®-M4 ve FPU çekirdekli 1-MB Flash belleğe ve 192-KB RAM'e sahip STM32F407VGT6 mikrodeneleyici
- USB OTG FS
- ST MEMS 3 eksenli ivmeölçer
- ST-MEMS ses sensörü, çevresel dijital mikrofon
- Entegre sınıf D hoparlör sürücülü ses DAC'ı
- Kullanıcı ve reset düğmeleri
- Sekiz adet LED
- USB iletişimi için LD1 (kırmızı/yeşil)
- 3.3 V güç açık olduğunda LD2 (kırmızı)
- Dört kullanıcı LED'i, LD3 (turuncu), LD4 (yeşil), LD5 (kırmızı) ve LD6 (mavi)
- İki USB OTG LED'i, LD7 (yeşil) VBUS ve LD8 (kırmızı) aşırı akım
- Micro-AB ile USB Stereo kulaklık çıkış jakı
- Prototipleme kartına hızlı bağlantı ve kolay sondalama için tüm LQFP100 I/O'ları için 2.54 mm pitch uzatma başlığı

- ST-LINK, USB VBUS veya harici kaynaklar Harici uygulama güç kaynağı: 3 V ve 5 V
- STM32CubeF4 MCU Paketi'nin bir parçası olan çeşitli örnekleri içeren kapsamlı ücretsiz yazılım veya geleneksel standart kütüphaneleri kullanmak için STSW-STM32068 USB yeniden numaralandırma yetenekli ST-LINK/V2-A hata ayıklayıcı/programlayıcı ile donatılmış yerleşik

Bu geliştirme kartı üzerinde yerleşik olarak ST-LINK/V2-A bulunduğundan direkt olarak bilgisayar ile haberleşmesi sağlanabilmektedir. Böylece çalışma içerisinde harici bir programlayıcı kullanılmasına gerek görülmemiştir.

4.2. ESP32-WROOM-32DA

ESP32-WROOM-DA, iki yönlü PCB antenlere sahip güçlü bir Wi-Fi + Bluetooth + Bluetooth LE MCU modülüdür. Bu modül, ESP32-WROOM-32E ile aynı pin düzenine sahiptir, ancak bazı pinler kullanılmamıştır (Şekil 4.2). Bu da bu iki modül arasında hızlı ve kolay bir geçişi kolaylaştırır. Tek bir modülde iki benzersiz anten tasarımına sahip olan ESP32-WROOM-DA, geniş bir spektrum üzerinde istikrarlı bağlantıya ihtiyaç duyan IoT uygulamalarının geliştirilmesi için kullanılabilir veya zorlu ve tehlikeli ortamlarda Wi-Fi dağıtımı yapılabilir veya Wi-Fi kapsama alanında iletişim sorunlarının üstesinden gelmesine yardımcı olabilir. Bu modül, akıllı ev, endüstriyel kontrol, tüketici elektroniği vb. için iç mekân ve dış mekan cihazları için ideal bir seçimdir.



Şekil 4.2. ESP32-WROOM-DA Mikroişlemcisi

Modülün çekirdeğinde ESP32-D0WD-V3 çipi bulunur. Gömülü çip ölçeklenebilir ve uyarlanabilir olarak tasarlanmıştır. İki adet ayrı olarak kontrol edilebilen CPU çekirdeği bulunmakta ve CPU saat frekansı 80 MHz ila 240 MHz arasında ayarlanabilir. Çip ayrıca, çevre birimlerinin izlenmesi gibi yüksek hesaplama gücü gerektirmeyen görevleri yerine getirirken güç tasarrufu sağlamak için CPU yerine kullanılabilen düşük güçlü bir yardımcı işlemciye de sahiptir. ESP32-D0WD-V3, kapasitif dokunmatik sensörlerden SD kart arayüzüne, Ethernet'e, yüksek hızlı SPI, UART, I2S ve I2C gibi zengin bir yelpazede birleşik bir çevre birimlerini içerir (Espressif, 2023).

4.3. NVIDIA Jetson Nano

Jetson Nano, modern yapay zeka teknolojisinin gücünü her yerde kullanıma sunan bir platformdur. Görüntü sınıflandırma, nesne algılama, segmentasyon ve konuşma işleme gibi uygulamalar için birden fazla nöral ağı paralel olarak çalıştırabilme yeteneği ile küçük ve güçlü bir bilgisayardır (Şekil 4.3). Bu platform, yalnızca 5 watt gibi düşük bir güç tüketimiyle çalışırken kullanıcı dostu bir arayüz sunar.



Şekil 4.3. Nvidia Jetson Nano Geliştirme Kartı (Nvidia, 2022).

Güç sınırlı ortamlarda kullanım için tasarlanan Jetson Nano, endüstri lideri hesaplama yeteneklerini, 64-bit işletme yeteneklerini ve entegre gelişmiş çok işlevli ses, video ve görüntü işleme hatlarını 260 pinli bir SODIMM'e sıkıştırır. Maxwell GPU

mimarisi, tüketilen her watt başına maksimum performans elde etmek amacıyla birkaç mimari iyileştirme içerir. Jetson Nano serisi modülün temel bileşenleri şunlardır:

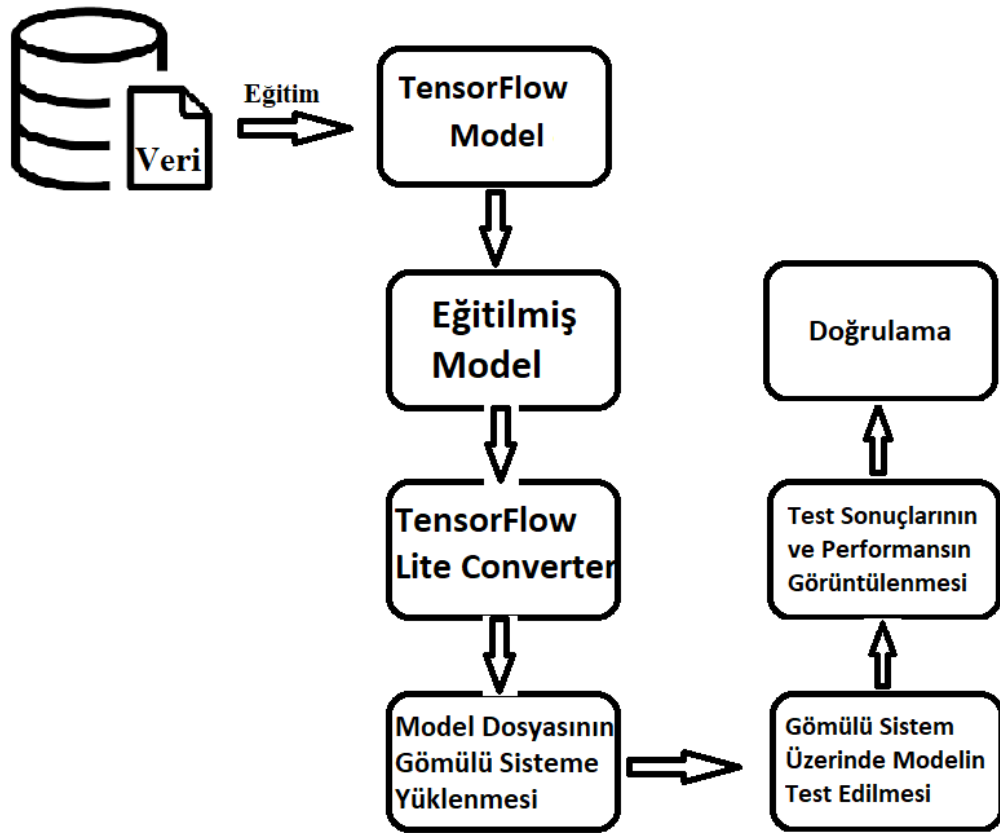
- NVIDIA® Tegra® X1 series SoC
 - NVIDIA Maxwell GPU
 - ARM® quad-core Cortex®-A57 CPU Complex
- 4GB LPDDR4 bellek
- 16GB eMMC 5.1 hafıza
- Gigabit Ethernet (10/100/1000 Mbps)
- PMIC, regülatörler, güç ve voltaj monitörleri
- 260-pin anahtarlı konektör (hem yüksek hızlı hem de düşük hızlı endüstri standardını gösterir I/O)
- Çip üstü sıcaklık sensörleri (Nvidia, 2022).

5. GÖMÜLÜ SİSTEM ÜZERİNDE NESNE TESPİTİ

Gömülü sistem üzerinde nesne tespiti yapabilmek gerçek zamanlı işlem sağlanabilmesi adına oldukça önemlidir (Iguernaissi ve ark., 2019). Bu yüzden ki gömülü bir nesne tespit sistemi tasarlariken dikkate alınması gereken iki önemli faktör, algoritmaların gerçek zamanlı performansı ve sağlamlığıdır. Senaryo karmaşıklığı nedeniyle, nesne takip algoritmaları hala çeşitli zorluklarla karşı karşıyadır (Zhang ve ark., 1987; Wu ve ark., 2015).

Gömülü sistemin nesne tespiti görevi, algoritmanın gerçek zamanlı ve sağlam olmasını gerektirir. Öğrenme ve karmaşık senaryolardaki tespit algoritmasının düşük sağlamlığı gibi sorunları çözmek amacıyla, gömülü sistemler için hızlı ve doğru bir gerçek zamanlı tespit algoritması uygulamak gerekmektedir. (Ji ve ark., 2021).

Tez çalışmasında gömülü sistem üzerinde yapılacak olan işlemlerin blok şeması aşağıdaki Şekil 5.1’de gösterilmiştir.



Şekil 5.1. Uygulamanın Gömülü Sisteme Dayalı Blok Şeması

5.1. Derin Öğrenmeye Genel Bakış

Yapay zeka, özellikle derin öğrenme, son zamanlarda algılama görevlerinde önemli adımlar atarak makinelerin karmaşık verileri daha iyi temsil etmesine ve yorumlamasına olanak tanımaktadır (Kalmet ve ark., 2020). Derin Öğrenme, girdiyi çıktıya eşlemek için matematiksel işlevleri kullanan makine öğreniminin daha da geniş bakarsak yapay zeka alanının bir alt kümesidir. Bu işlevler, girdi ve çıktı arasında bir ilişki oluşturmalarını sağlayan verilerden gereksiz olmayan bilgi veya kalıpları çıkarabilir. Bu öğrenme olarak bilinir ve öğrenme sürecine eğitim denir.

Geleneksel bilgisayar programlamada, istenen çıktıyı elde etmek için girdi ve bir dizi kural bir araya getirilir. Makine öğreniminde ve derin öğrenmede, girdi ve çıktı kurallarla ilişkilendirilir (Şekil 5.2). Bu kurallar yeni girdilerle birleştirildiğinde istenen sonuçları verir (Barla, 2021).



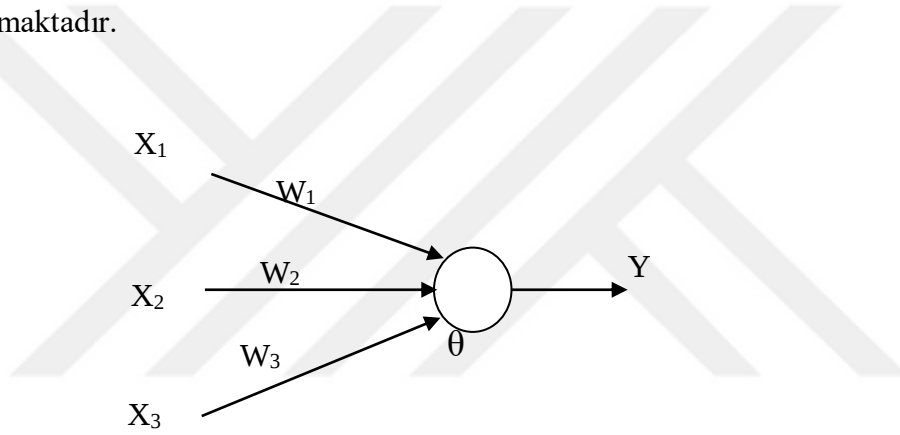
Şekil 5.2. Klasik Programlama / Derin Öğrenme

5.2. Yapay Sinir Ağları

Yapay sinir ağları (YSA), bilgisayar sistemleri olarak geliştirilmiştir ve insan beyninin özelliklerinden olan öğrenme yoluyla yeni bilgiler türetebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri otomatik olarak gerçekleştirmek için tasarlanmıştır. Beyindeki biyolojik sinir ağlarının yapısını, öğrenme, hatırlama ve genelleme kabiliyetlerini taklit eder (Şengöz, 2017). Yapay sinir ağlarında öğrenme işlemi örnekler kullanılarak gerçekleştirilir. Öğrenme esnasında giriş çıkış bilgileri verilerek, kurallar koyulur. Bu sistemler, insan müdahalesine ihtiyaç duymadan, bağımsız bir şekilde bu yetenekleri kullanabilmeyi hedeflemektedir (Kılıç, 2021).

Sinir ağının insan vücudunda önemli ve anahtar bir rol oynadığı biyoloji konusundan tanıtılmıştır. İnsan vücudunda çalışma, sinir ağı yardımıyla yapılır. Sinir Ağı, sayıları milyonlarca olan birbirine bağlı nöronlardan oluşan bir ağıdır. Bu birbirine bağlı nöronların yardımıyla insan vücudunda tüm paralel işlemler yapılır ve insan vücudu Paralel İşlemenin en iyi örneğidir. Bir nöron, bazı elektriksel ve kimyasal değişimlerin yardımıyla bir nörondan diğerine bilgi işleyen özel bir biyolojik hücredir.

Yapay sinir hücreleri de biyolojik sinir hücrelerine benzer yapıdadır. Yapay nöronlar da aralarında bağ kurarak yapay sinir ağlarını oluştururlar. Bu yapı şekil 5.4'te gösterilmiştir. Aynı biyolojik nöronlarda olduğu gibi yapay nöronların da giriş sinyallerini aldıkları, bu sinyalleri toplayıp işledikleri ve çıktıları ilettikleri bölümleri bulunmaktadır.



Şekil 5.3. Yapay sinir hücresinin yapısal gösterimi

Burada; X: Girdi, W: Ağırlık, θ : Birleştirme Fonksiyonu, Y: Çıkış olarak alınır. Yapay sinir ağları, tespit, sınıflandırma, tahmin, kontrol, veri ilişkilendirme, veri filtreleme ve yorumlama gibi birçok alanda kullanılmaktadır. Hangi problemin hangi ağla daha iyi çözülebileceğini belirlemek için ağların özelliklerini ve problemlerin özelliklerini karşılaştırmak gerekmektedir (Yıldırım, 2020).

5.3. Derin Öğrenme Algoritmaları

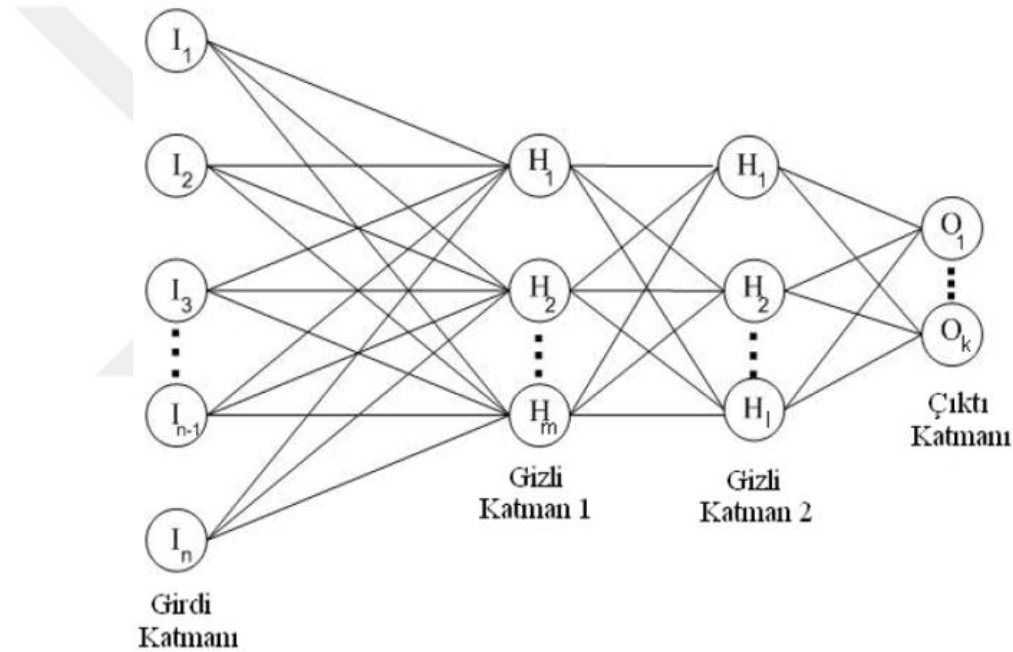
Veri tahmini modellemesi ve istatistik altında veri odaklı tekniklere dayalı bir veri bilimi unsuru olan çok önemli bir alandır. Böyle bir uyum sağlama, öğrenme ve buna uygun şekilde işlev gösterme gibi insan benzeri bir yetenek için, popüler olarak algoritma olarak adlandırdığımız bazı güçlü unsurlar olmalıdır.

Derin öğrenme algoritmaları, önceden eğitilmiş bir görevi yerine getirmek üzere eğitilen bir dizi karar verme ağının birkaç katmanından geçirilerek dinamik olarak

oluşturulur. Daha sonra, her biri basit katmanlı temsillerden geçirilir ve bir sonraki katmana geçer. Ancak, çoğu makine öğrenimi, yüzlerce özellik veya sütunla uğraşan veri kümeleri üzerinde oldukça iyi çalışmak üzere eğitilmiştir (Ülker ve İnık, 2017).

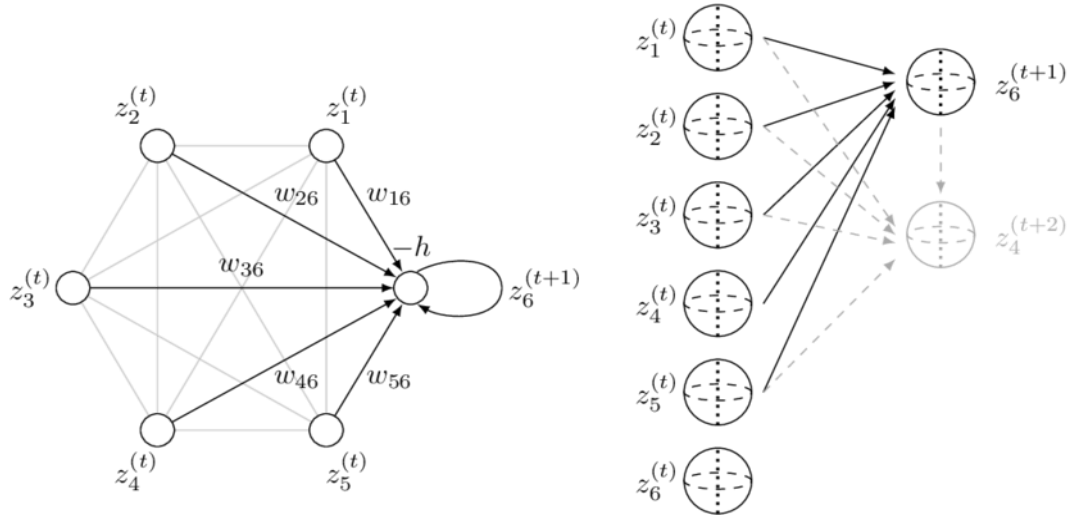
5.3.1. Evrişimsel Sinir Ağları

Evrişimsel sinir ağları (CNN - Convolutional neural networks), girdi olarak bir görüntü alarak, derin öğrenme algoritması olarak görüntüdeki farklı görünüşleri veya nesneleri birbirinden ayırabilen bir yapıdır. Bu tür sinir ağları, temel olarak görüntülerin sınıflandırılması (örneğin, gördüklerinin adlandırılması), benzerliklere göre kümeleme (fotoğraf arama) ve bazı senaryolarda nesneleri tanımlama gibi görevler için kullanılan derin yapay sinir ağlarıdır (Okumuş, 2019).



Şekil 5.4. Evrişimli Sinir Ağı modeli

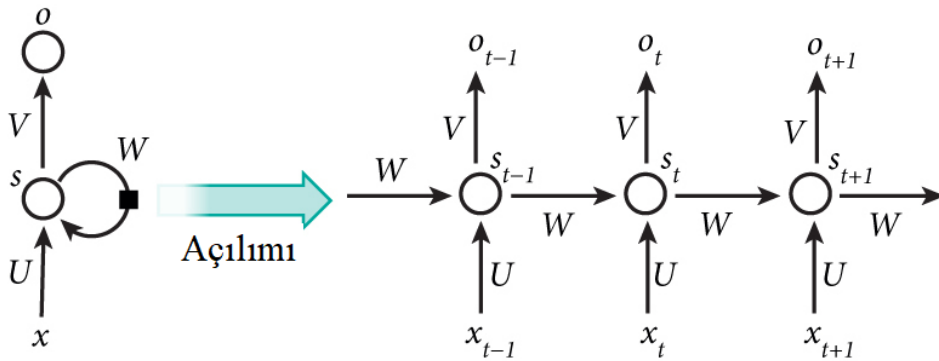
Derin öğrenme algoritmaları, önceden eğitilmiş bir görevi yerine getirmek üzere eğitilen bir dizi karar verme ağının birkaç katmanından geçirilerek dinamik olarak oluşturulur. Daha sonra, her biri basit katmanlı temsillerden geçirilir ve bir sonraki katmana geçer. Ancak, çoğu makine öğrenimi, yüzlerce özellik veya sütunla uğraşan veri kümeleri üzerinde oldukça iyi çalışmak üzere eğitilmiştir (Karakuş, 2018). Örnek olarak Hopfield ağı modeli ve Kohonen özellik haritası verilebilir. Hopfield modeli, her biri birbirine bağlı sinir hücrelerinden oluşan bir kümeyi içerir. Girdi ve çıktı hücreleri arasında fark gözetilmez (Sezgin ve ark., 2012).



Şekil 5.5. Bir Hopfield ağı modelini güncelleme adımının şemaları

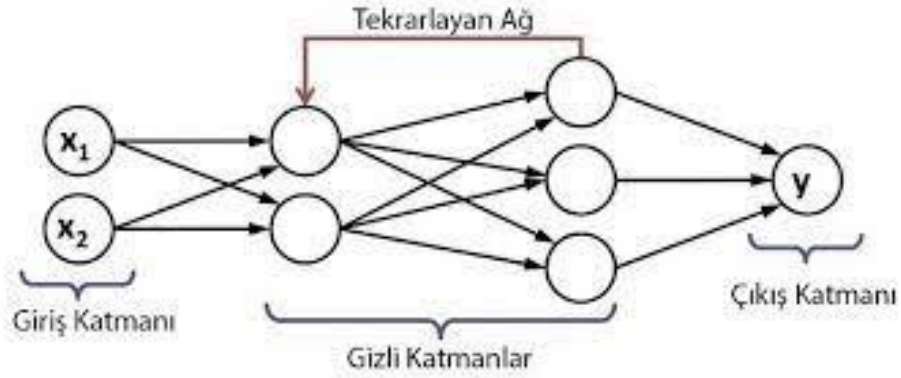
5.3.2. Tekrarlayan Sinir Ağları

Tekrarlayan sinir ağları (Recurrent Neural Network, RNN), yapay sinir ağları sınıfında düğümler arasında yönlendirilmiş bir döngü oluşturan bir modeldir. Bu sayede dinamik zamansal davranışları sergileyebilirler. İleri beslemeli sinir ağlarından farklı olarak, RNN'ler girdilerin ardışık dizilerini işlemek için kendi dahili belleklerini kullanabilirler (Schmidhuber, 1992). Tekrarlayan sinir ağlarının temel amacı ardışık bilgileri işlemektir. Geleneksel sinir ağlarında tüm girişlerin ve çıktılarının birbirinden bağımsız olduğu varsayılır. RNN'ler "tekrarlayan" olarak adlandırılır, çünkü her bir dizinin ögesi için aynı görevi tekrarlarlar ve çıktılar önceki hesaplamalara bağlıdır. RNN'leri düşünmenin başka bir yolu, bugüne kadar hesaplanan bilgilerle ilgili "bellek" taşıdıklarıdır. Teorik olarak, RNN'ler keyfi uzunluktaki dizilerdeki bilgileri kullanabilirler, ancak pratikte geriye sadece birkaç adım dönme yetenekleri vardır (Şeker ve ark., 2017).



Şekil 5.6. Tipik Bir RNN Örneği

Tekrarlayan sinir ağları, genellikle dil çevirisi gibi arka arkaya gelen bir dizi yapıda, bir sonraki noktanın tahmin edilmesi işlemini gerçekleştirmek için kullanılır. Bu yapı aşağıdaki şekil 5.8’da gösterilmiştir (Doğan ve Türkoğlu, 2018).

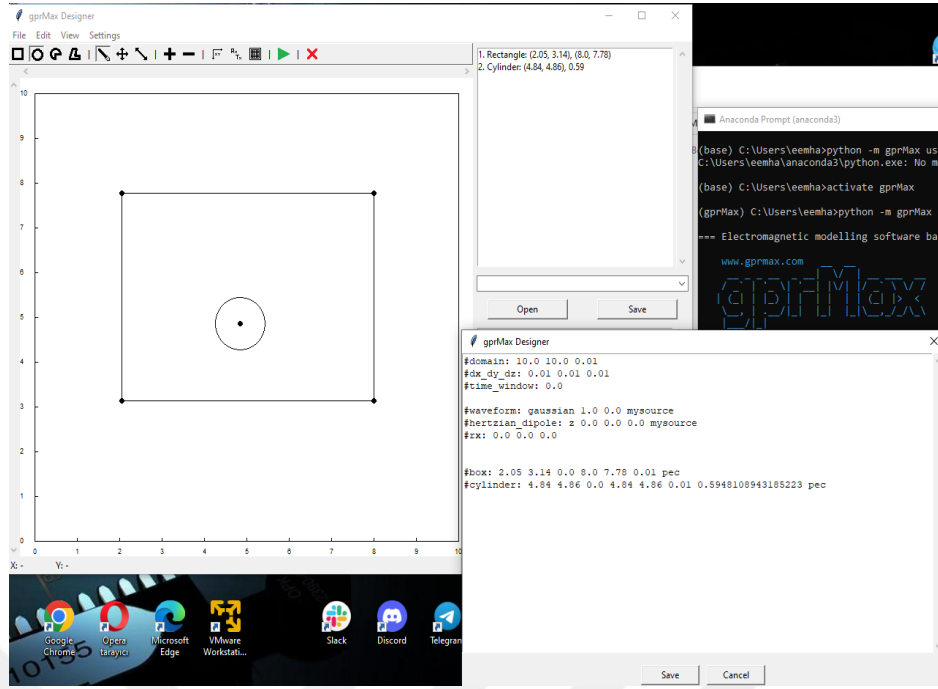


Şekil 5.7. Tekrarlayan Ağ Yapısı (Doğan ve Türkoğlu, 2018)

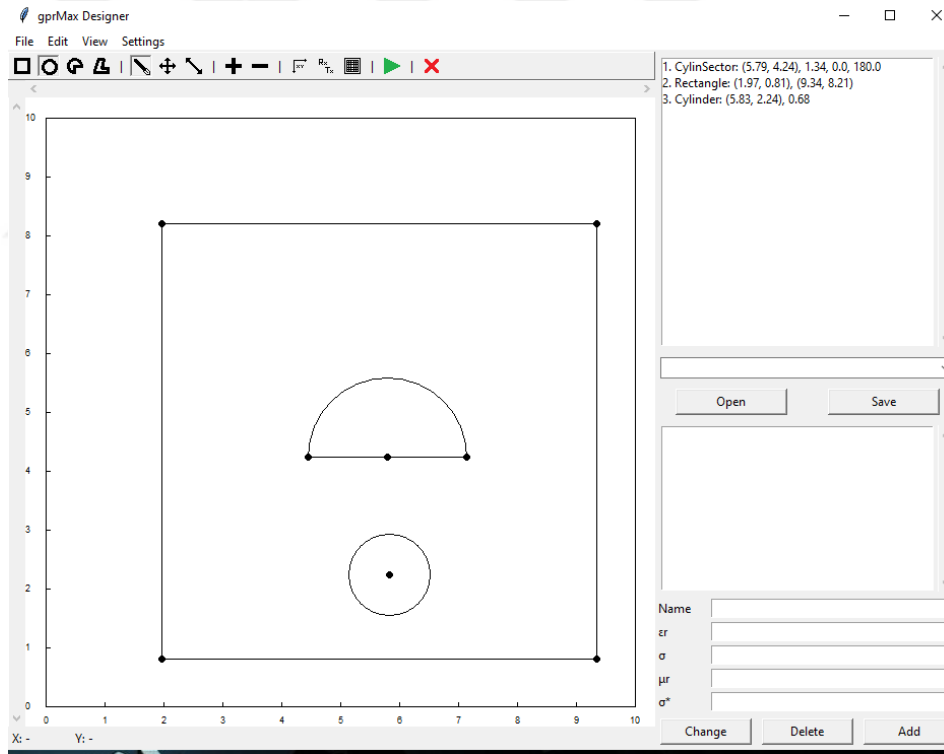
5.4. GprMax ile Yere Nüfuz Eden Radar Verisinin Elde Edilmesi

GprMax verilerinin elde edilmesi için açık kaynak koda sahip GprMax Designer programı ile model tasarlanmıştır. Bu program, Kraków'daki AGH Bilim ve Teknoloji Üniversitesi, Jeoloji, Jeofizik ve Çevre Koruma Fakültesi'nde bir yüksek lisans tezi olarak geliştirilmektedir (Siwek, 2019). Uygulamanın kurulumuna <https://github.com/tomsiwek/gprMax-Designer>, adresinden ulaşılabilir. Uygulama Python tabanlı olup Anaconda ortamında çalışabilmektedir. Gerekli kurulumlar yapıldıktan sonra Anaconda Prompt üzerinden “*conda activate gprMaxDesigner*” komutu ile uygulama çalıştırılmaktadır. Aşağıdaki şekil 5.8’da uygulamanın arayüzü gösterilmiştir.

Uygulama üzerinde model için çizilecek dört şekil bulunmaktadır. Bunlar; kutular, silindirler, silindirik sektörler ve çokgenlerdir. Aşağıdaki şekil 5.9’da tasarlanmış model gösterilmektedir.



Şekil 5.8. gprMax Designer’de model tasarımı



Şekil 5.9. gprMax Designer’de model tasarımı-2

Settings ana menüdeki öge, tarama tarama parametrelerinin yanı sıra eksen çizgileri aralığı, model boyutu, alan ayrıklaştırması gibi ekran ve model ayarlarının yapılmasına olanak tanır. Yine settings menüsünden kararlılık faktörü, dalga formu ve

genlik değerlerini ayarlamak, A ve B tarama seçeneği için olanaklar verilmiştir. Bu ayarlar Şekil 5.10 ve 5.11’de gösterilmiştir.

time stability factor:	1.0	time window [s]:	6.67e-08
waveform:	gaussian	frequency [Hz]:	9.99e+08
amplitude:	1.0		
source:	hertzian_dipole		
x [m]:	0.0	y [m]:	0.0
receiver			
x [m]:	0.0	y [m]:	0.0
survey type:	☒ Ascan	☐ rx array	☐ Bscan
src step			
Δx [m]:	0.0	Δy [m]:	0.0
rx step			
Δx [m]:	0.0	Δy [m]:	0.0
rx array			
max x [m]:	0.0	max y [m]:	0.0
messages:	yes	vtk file:	
geometry view:	no	snapshot time:	0.0
snapshot:	no		
snapshot file:			

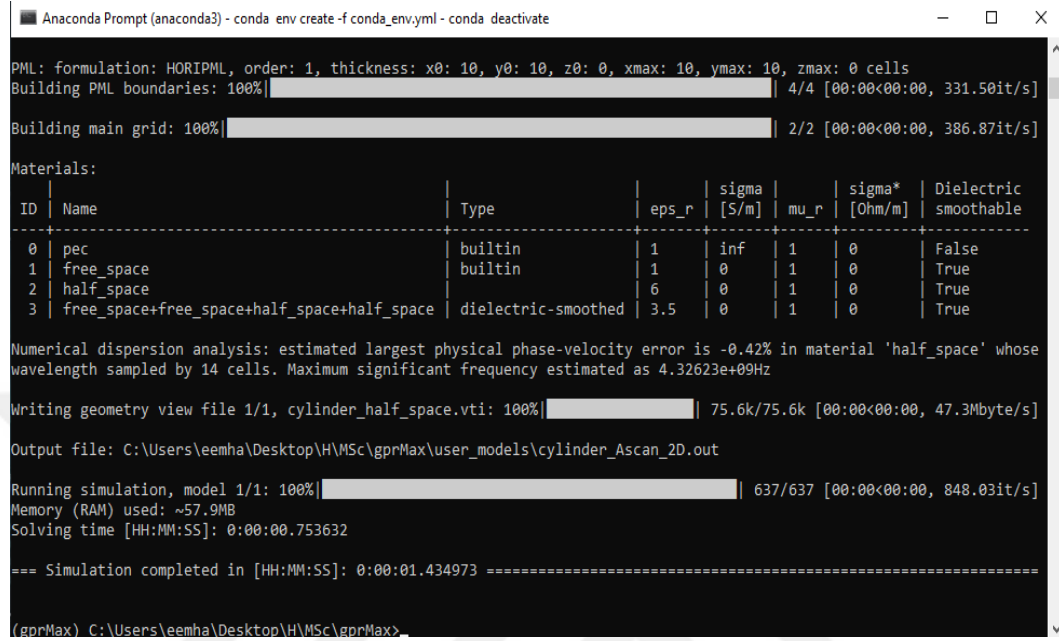
Şekil 5.10. gprMax Designer Üzerinde Modelin Ayar Arayüzü (A-Tarama)

time stability factor:	1.0	time window [s]:	3e-09
waveform:	ricker	frequency [Hz]:	1.5e+09
amplitude:	1.0		
source:	hertzian_dipole		
x [m]:	0.0	y [m]:	0.0
receiver			
x [m]:	0.08	y [m]:	0.17
survey type:	☐ Ascan	☐ rx array	☒ Bscan
src step			
Δx [m]:	0.002	Δy [m]:	0.0
rx step			
Δx [m]:	0.002	Δy [m]:	0.0
rx array			
max x [m]:	0.0	max y [m]:	0.0
messages:	yes	vtk file:	
geometry view:	no	snapshot time:	0.0
snapshot:	no		
snapshot file:			

Şekil 5.11. gprMax Designer Üzerinde Modelin Ayar Arayüzü (B-Tarama)

Oluşturulan modeli ayrıştırmak için, sonuç dosyasını kaydetmek gerekmektedir. Parse to gprMax araç çubuğundaki düğmesine tıklanarak ya da File\Parse to gprMax

ana menüden öğesini kullanarak dosya gprMax simülasyonu için çıktı sağlanmaktadır. Sonrasında gprMax simülasyonunu çalıştırılır. Bir A-taraması için modelin işlenmesi aşağıdaki Şekil 5.12’de gösterilmiştir.



```

Anaconda Prompt (anaconda3) - conda env create -f conda_env.yml - conda deactivate

PML: formulation: HORIPML, order: 1, thickness: x0: 10, y0: 10, z0: 0, xmax: 10, ymax: 10, zmax: 0 cells
Building PML boundaries: 100%| 4/4 [00:00<00:00, 331.50it/s]
Building main grid: 100%| 2/2 [00:00<00:00, 386.87it/s]

Materials:


| ID | Name                                        | Type                | eps_r | sigma [S/m] | mu_r | sigma* [Ohm/m] | Dielectric smoothable |
|----|---------------------------------------------|---------------------|-------|-------------|------|----------------|-----------------------|
| 0  | pec                                         | builtin             | 1     | inf         | 1    | 0              | False                 |
| 1  | free_space                                  | builtin             | 1     | 0           | 1    | 0              | True                  |
| 2  | half_space                                  |                     | 6     | 0           | 1    | 0              | True                  |
| 3  | free_space+free_space+half_space+half_space | dielectric-smoothed | 3.5   | 0           | 1    | 0              | True                  |



Numerical dispersion analysis: estimated largest physical phase-velocity error is -0.42% in material 'half_space' whose wavelength sampled by 14 cells. Maximum significant frequency estimated as 4.32623e+09Hz

Writing geometry view file 1/1, cylinder_half_space.vti: 100%| 75.6k/75.6k [00:00<00:00, 47.3Mbyte/s]

Output file: C:\Users\eeemha\Desktop\H\MSc\gprMax\user_models\cylinder_Ascan_2D.out

Running simulation, model 1/1: 100%| 637/637 [00:00<00:00, 848.03it/s]
Memory (RAM) used: ~57.9MB
Solving time [HH:MM:SS]: 0:00:00.753632

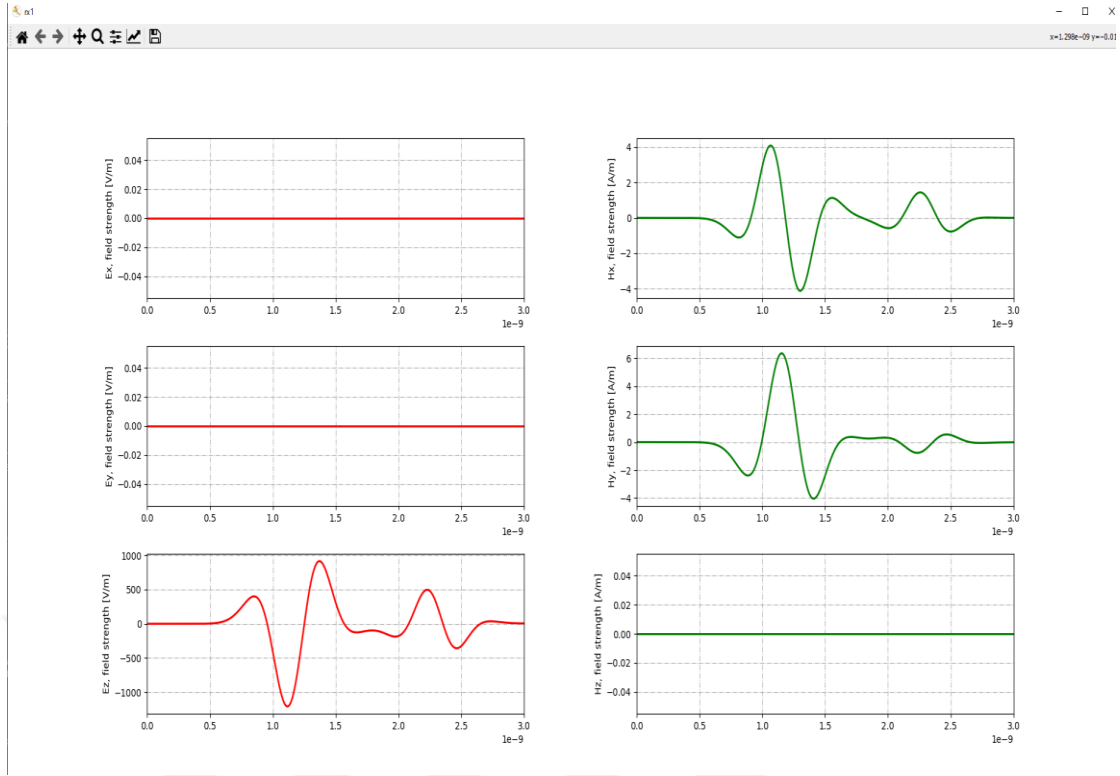
=== Simulation completed in [HH:MM:SS]: 0:00:01.434973 =====
(gprMax) C:\Users\eeemha\Desktop\H\MSc\gprMax>

```

Şekil 5.12. gprMax ile A Tarama Modelinin İşlenmesi

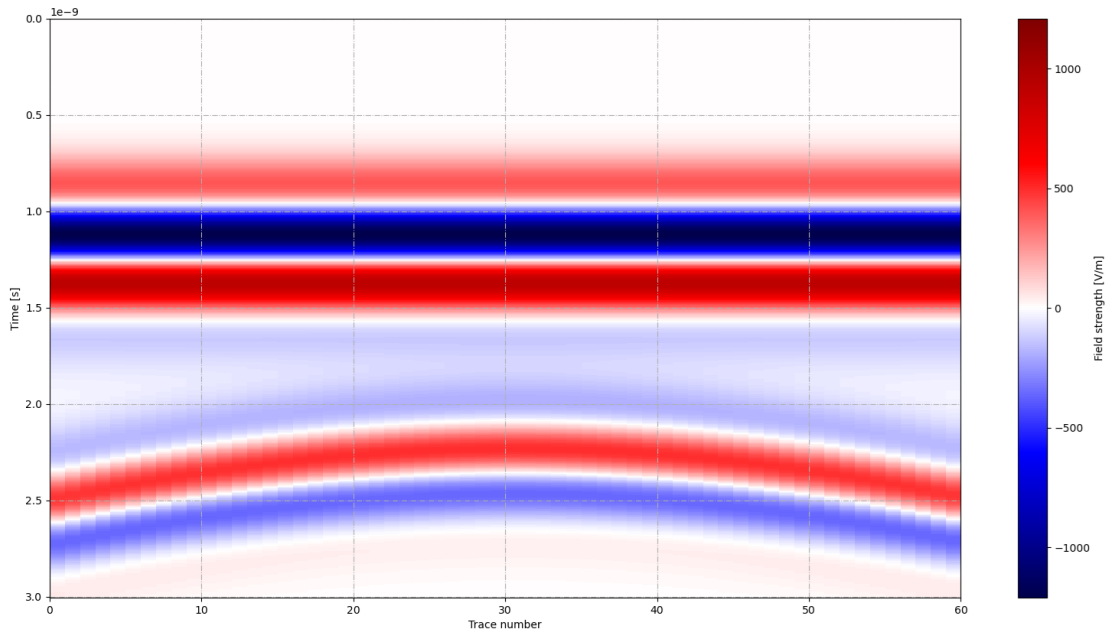
Modelin gprMax programı üzerinden tanınması üzerine plot komutu ile çizdirilip simüle edilmesi gerekmektedir. Bu komut:

“-m tools.plot_Ascan_user_models/cylinder_Ascan_2D.out” şeklindedir. İşlenen modele ait simülasyon aşağıdaki Şekil 5.13’te gösterilmiştir.



Şekil 5.13. gprMax ile A-Tarama verilerinin Simülasyonu

B-taramada yapılacak işlemler, aynı şekilde A-taramadaki gibi yapılır. Şekil 5.16'da B-tarama verisinin simülasyonu gösterilmiştir (Ez alan bileşeni). Yine, sinyalin ilk kısmı ($\sim 0.5-1.5$ ns), vericiden alıcıya doğrudan dalgayı temsil eder. Ardından, hiperbolik şekli oluşturan metal silindirden yansıyan dalga ($\sim 2-3$ ns) gelir.



Şekil 5.16. gprMax ile B-Tarama Verilerinin Simülasyonu

Bu çıktıların gprMax üzerinde oluşmasını sağlayan girdi dosyasına ait parametreler aşağıdaki gibidir.

```

1 #title: B-scan from Thesis_Model_cylinder in a dielectric half-space
2 #domain: 0.240 0.210 0.002
3 #dx_dy_dz: 0.002 0.002 0.002
4 #time_window: 3e-9
5
6 #material: 6 0 1 0 half_space
7
8 #waveform: ricker 1 1.5e9 my_ricker
9 #hertzian_dipole: z 0.040 0.170 0 my_ricker
10#rx: 0.080 0.170 0
11#src_steps: 0.002 0 0
12#rx_steps: 0.002 0 0
13
14#box: 0 0 0 0.240 0.170 0.002 half_space
15#cylinder: 0.120 0.080 0 0.120 0.080 0.002 0.010 pec

```

```

1 #title: B-scan from a metal cylinder buried in a dielectric half-space
2 #domain: 0.24 0.21 0.002
3 #dx_dy_dz: 0.002 0.002 0.002
4 #time_window: 3e-09
5
6 #waveform: ricker 1.0 1500000000.0 mysource
7 #hertzian_dipole: z 0.0 0.0 0.0 mysource
8 #rx: 0.08 0.17 0.0
9 #src_steps: 0.002 0.0 0.0
10#rx_steps: 0.002 0.0 0.0
11
12#material: 6.0 0.0 1.0 0.0 half_space
13
14#box: 0.0 0.0 0.0 0.24 0.17 0.002 half_space
15#cylinder: 0.118 0.04 0.0 0.118 0.04 0.002 0.01 pec
16#cylindrical_sector: z 0.118 0.086 0.0 0.002 0.044 0.0 180.0 pec

```

Ölçümlerde kullanılan Maxwell's Denklemleri aşağıda verilmiştir.

$$\nabla \times H = \sigma E + \epsilon * \frac{\partial E}{\partial t}$$

$$\nabla \times E = -\frac{\partial B}{\partial t}$$

$$\nabla \cdot D = \rho \quad D = \epsilon E$$

$$\nabla \cdot B = 0 \quad B = \mu H$$

(5.1)

Ölçümlerde kullanılan Zaman Alanı Denklemleri aşağıda verilmiştir.

$$\nabla \times H = \sigma E + \epsilon \frac{\partial E}{\partial t} \quad (5.2)$$

Bu denklem Faraday Kanunu'nu temsil eder ve manyetik alanın (H) dönmesini elektrik alanla (E) ve elektrik alanının zaman içindeki değişim oranı ($\partial E/\partial t$) ile ilişkilendirir.

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad (5.3)$$

Bu denklemler, frekans etkilerini göz önünde bulundurmadan zaman alanında alan davranışını açıklar. Bu denklem Faraday Kanunu'nun bir formudur ve elektrik alanın dönmesini negatif zaman değişim oranıyla (B) ilişkilendirir.

Ölçümlerde kullanılan Frekans Alanı Denklemleri aşağıda verilmiştir. Bu denklemler, frekans alanında alanları analiz etmek ve farklı frekansta elektromanyetik cevapları incelemek için kullanılır. Bu denklem Faraday Kanunu'nu frekans alanına genişletir, iletkenlik (σ) ve kompleks izin verme (ϵ_ω) etkilerini hesaba katar. Frekans alanında bu denklem, elektrik alanın dönmesini negatif açısal frekans (ω) ile manyetik alan (B_ω) çarpımına bağlar.

$$\begin{aligned} \nabla \times H_\omega &= \sigma E_\omega + \epsilon_\omega j\omega E_\omega \\ \nabla \times E &= -j\omega B_\omega \end{aligned} \quad (5.4)$$

Ölçümlerde kullanılan Kompleks Dielektrik Özellikleri aşağıdaki gibidir.

$$\epsilon_c = \epsilon_\omega - j\frac{\sigma}{\omega} \epsilon_\omega = \epsilon_\infty + x_\omega \quad (5.5)$$

Bu denklemler, malzemelerin kompleks dielektrik özelliklerini açıklar ve elektromanyetik alanlara nasıl tepki verdiklerini belirler.

Ölçümlerde kullanılan Çoklu Debye (Multi Debye) denklemi aşağıda verilmiştir.

$$x_\omega = \sum_{p=1}^p \frac{\epsilon_{s,p} - \epsilon_\infty}{1 + j\omega t_{0,p}} \quad (5.6)$$

Burada x_ω , gevşeme sürelerinin ($t_{0,p}$) toplamını ve statik ile arka plan izin verme değerleri arasındaki farkı içeren bir toplam olarak hesaplanır. Ölçümlerde kullanılan Çoklu Lorentz (Multi Lorentz) aşağıdaki gibidir.

$$x_{\omega} = \sum_{p=1}^p \frac{(\epsilon_{s,p} - \epsilon_{\infty})\omega_p^2}{\omega_p^2 + 2j\omega\delta_p - \omega^2} \quad (5.7)$$

Burada x_{ω} , birden çok Lorentzian gevşeme sürecinin rezonans frekanslarını (ω_p) ve sönüm faktörlerini (δ_p) içerir. Ölçümlerde kullanılan Havriliak-Negami:

$$x_{\omega} = \frac{\epsilon_{s,p} - \epsilon_{\infty}}{((1 + (j\omega t_{0,p}))^a)^b} \quad (5.8)$$

Burada x_{ω} , gevşeme sürelerine ($t_{0,p}$) ve dayalı genelleştirilmiş bir gevşeme sürecini temsil eder. Ölçümlerde kullanılan Çoklu Drude (Multi Drude) denklemi aşağıda verilmiştir.

$$x_{\omega} = \sum_{p=1}^p \frac{(\omega_p/V_{c,p})^2}{1 + j\frac{\omega}{V_{c,p}}} + \frac{1}{j\omega} \sum_{p=1}^p \frac{\omega_p^2}{V_{c,p}} \quad (5.9)$$

Burada x_{ω} , her gevşeme süreci için Drude benzeri terimleri içerir, rezonans frekansları (ω_p) ve çarpışma frekansları ($V_{c,p}$) içerir. Ölçümlerde kullanılan Jonscher denklemi aşağıda verilmiştir.

$$x_{\omega} = \sum_{p=1}^p A_p \left(-\frac{\omega}{\omega_p}\right)^{n_p-1} \quad (5.10)$$

Burada x_{ω} , iletkenliği tanımlamak için parametreler (A_p , ω_p , n_p) kullanarak hesaplanır. Ölçümlerde kullanılan Debye Gevşemesi (debye Relaxation) denklemi aşağıda verilmiştir.

$$\epsilon = \epsilon_{\infty} + \frac{\epsilon_s - \epsilon_{\infty}}{1 + j\omega t_0} \quad (5.11)$$

Burada ϵ , gevşeme davranışı sergileyen bir malzemenin kompleks izin vermesini temsil eder, statik (ϵ_{∞}) ve doyumluk (ϵ_s) izin verme değerleri ve gevşeme süresi (t_0) ile karakterizedir.

Bu denklemler, elektromanyetik olayları anlamada ve özellikle malzemelerin farklı frekansta elektrik ve manyetik alanlara tepkilerini incelemekte temel bir rol oynar.

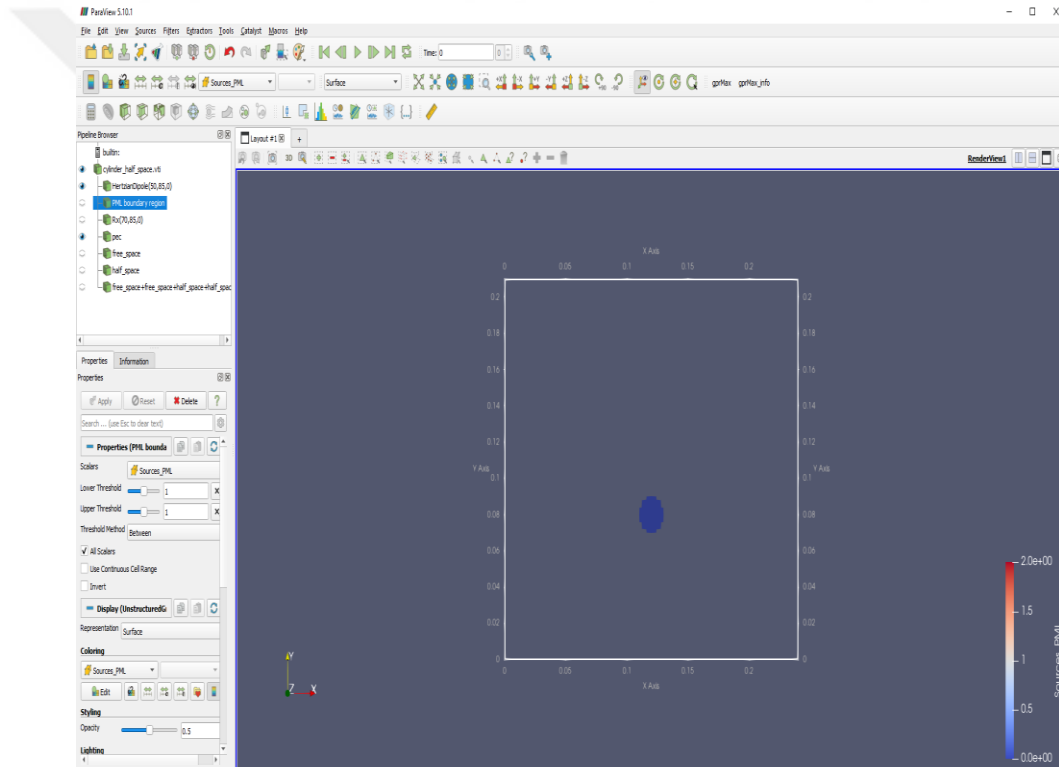
5.5. GprMax Çıktısının Paraview Programı Üzerinde Görüntülenmesi

Paraview, bilimsel veri görselleştirme ve analizinde kullanılan bir programdır. GprMax tarafından üretilen sonuçlar, Paraview aracılığıyla 2D-3D görselleştirme,

kesitler, renklendirme ve diğer görsel analiz teknikleri ile gösterilmiştir. Bu, GprMax tarafından simüle edilen elektromanyetik dalga yayılımının, gerçekçi ve anlaşılır bir görselleştirmesini sağlar.

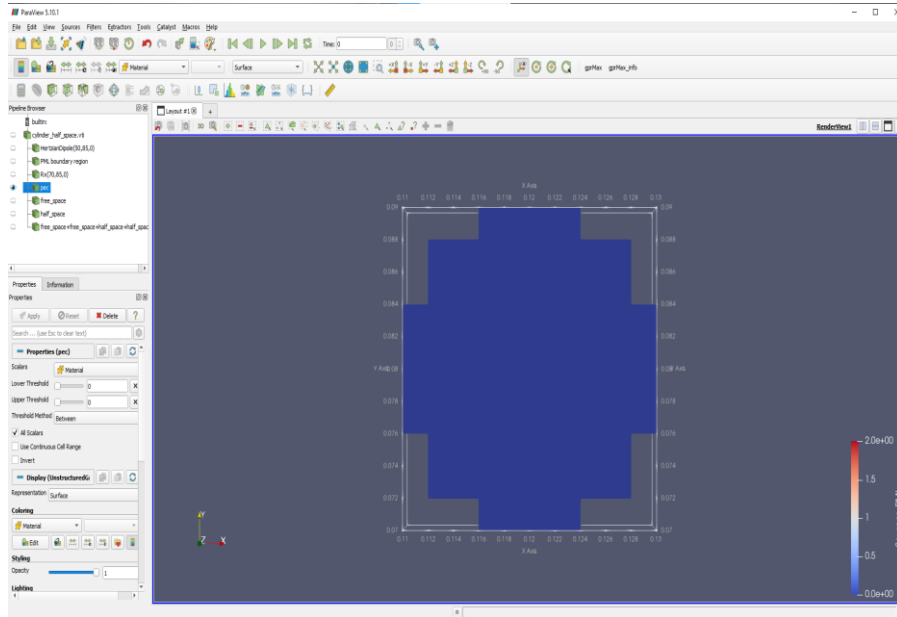
Ayrıca, Paraview'un sunduğu çoklu görüntüleme özellikleri sayesinde, GprMax çıktısının farklı perspektiflerden incelenmesi ve analiz edilmesi mümkündür (Savita ve Pallavi, 2022).

Tez çalışmasında GprMax tarafından üretilen sonuçların Paraview programı üzerinde görselleştirilerek nasıl analiz edildiği detaylı bir şekilde adım adım gösterilmiştir. İlk adım olarak gprMax çıktısı paraview programı üzerinden alınıp aşağıdaki şekil 5.17'deki gibi görüntülenmiştir.



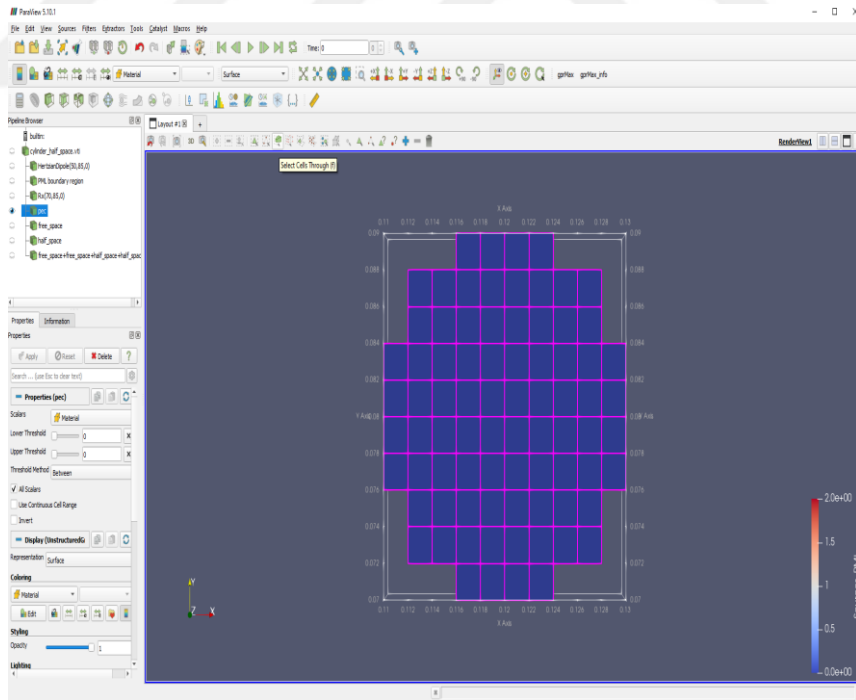
Şekil 5.17. gprMax ile B-Tarama Verilerinin Simülasyonu

Bu görünümünden veri alınabilmesi için pec görüntüsüne ihtiyaç duyulmaktadır. Böylelikle alanın yalnızca pec görüntüsü seçilmiştir. Bu durum aşağıdaki şekil 5.18'de gösterilmiştir.



Şekil 5.18. gprMax ile B-Tarama Verilerinin Simülasyonu

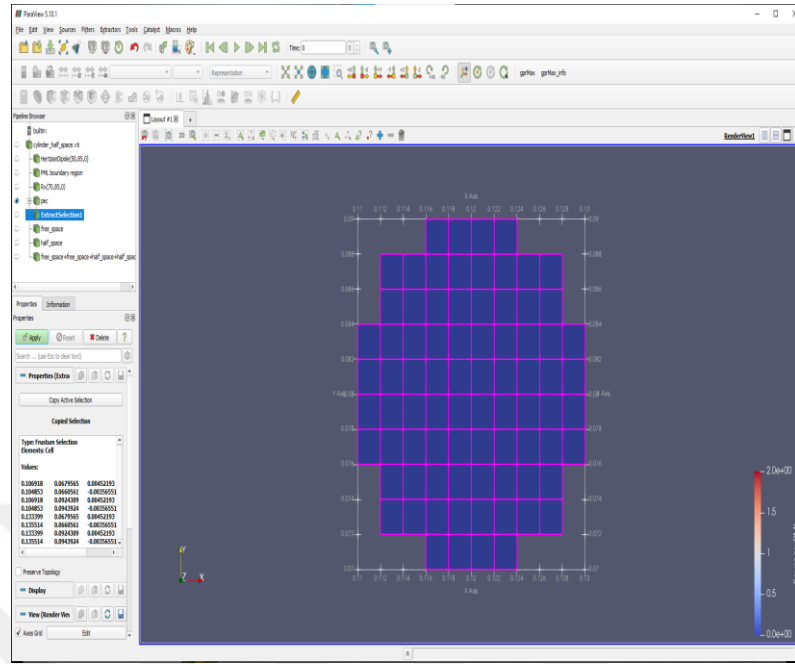
Program üzerinde verilerin hücreler halinde seçilmesi gerekmektedir. Bunun için program üzerinde “Select Cell Through” menüsü ile tüm tabla seçilmiştir. Bu seçim aşağıdaki Şekil 5.19’da gösterilmiştir.



Şekil 5.19. gprMax ile B-Tarama Verilerinin Simülasyonu

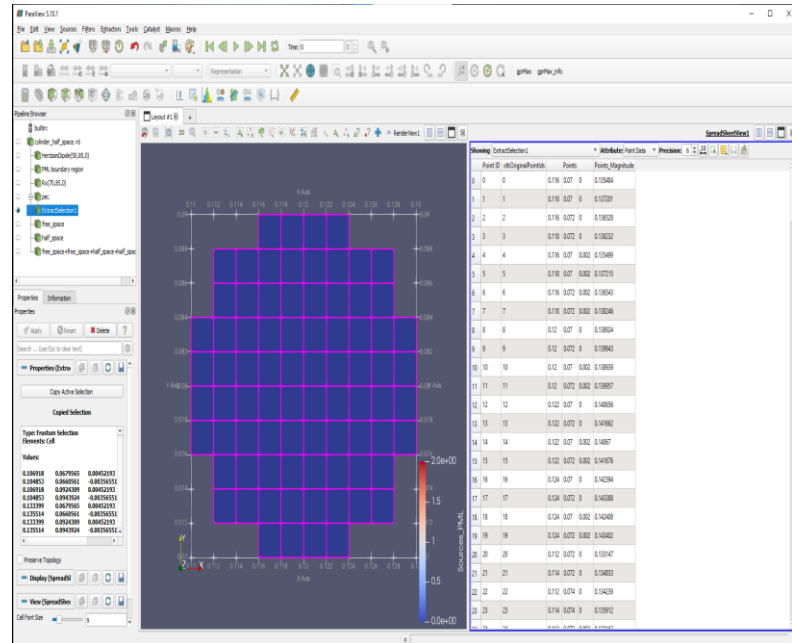
Bu işlem sonrasında seçilen bölüme ait değerler için ekstra bir alan oluşturulur bu alan uygulamada ExtractSelection1 olarak adlandırılmıştır. Bu alana ait değerler ise

ekranın sol kısmında yer alan kısımda yer almaktadır. Bu kısım aşağıdaki şekil 5.20’de gösterilmiştir.



Şekil 5.20. gprMax ile B-Tarama Verilerinin Simülasyonu

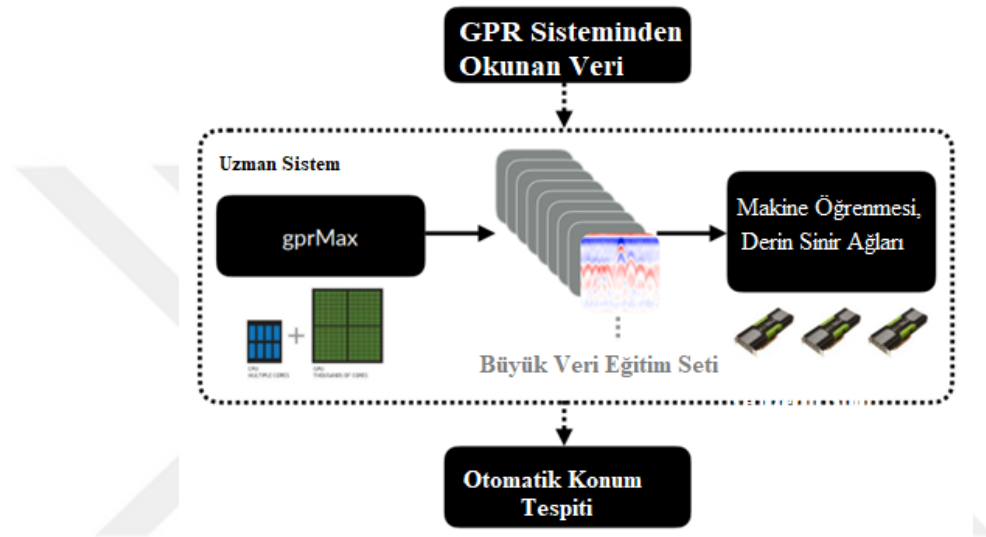
Programın sunmuş olduğu e tablo görünümü ile bu değerleri .csv, .xls çıktısı olarak almak mümkündür. Böylelikle veriler bir csv formatında bulunacaktır. Verilerin tablo halinde çekilmesini gösteren görsel aşağıdaki şekil 5.21’de gösterilmiştir. Böylelikle makine öğrenmesi için gerekli veriler elde edilmiştir.



Şekil 5.21. gprMax ile B-Tarama Verilerinin Simülasyonu

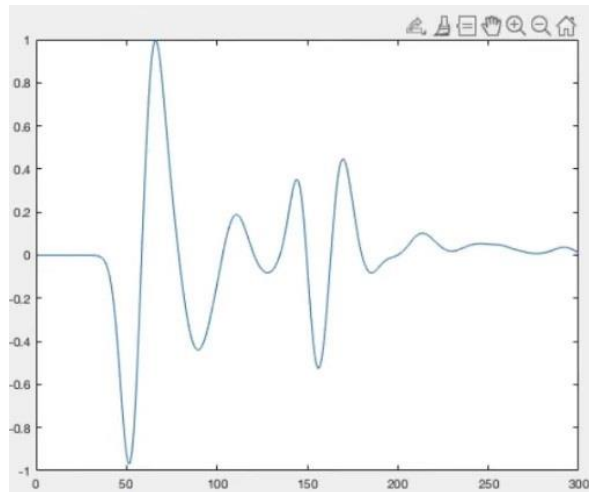
5.6. MATLAB Üzerinden Bilgisayar Tabanlı YNR ile Nesne Tespiti

YNR ile nesne tespiti gömülü sisteme taşımadan önce MATLAB ortamında test yapılması tez çalışması için daha verimli olacağı düşünülmüştür. Buna dayanarak Dr. Iraklis Giannakis'in yapmış olduğu MATLAB üzerinde nesne tespiti çalışmasından esinlenerek çalışma sağlanmıştır. Çalışma 11.nesil intel i7-11800H işlemcisi, 16GB RAM, NVIDIA GeForce RTX 3050 Ti ekran kartına sahip MSI marka notebook üzerinde yapılmıştır. Bu çalışmaya ait blok diyagram aşağıdaki Şekil 5.22'deki gibidir.



Şekil 5.22. gprMax ile Makine Öğrenmesi Kullanarak Nesne Tespiti Diyagramı (Giannakis ve ark., 2021)

Diyagrama bağlı kalarak gprMax ile elde edilen veriler MATLAB üzerine çekilmiştir. Bu veri setine ait kolonlar seçilerek o kolondaki sinyal çizdirilmiştir. Toplamda 300 sütun ve 1993 satırdan oluşmaktadır. Bir sütuna ait veriye ait grafiğin gösterimi Şekil 5.23'te verilmiştir.



Şekil 5.23. MATLAB Üzerinde Elde Edilen Veri Setine Ait Kolonun Çizdirilmesi

Bu veriler ışığında MATLAB üzerinde ileri beslemeli ağ kurulmuştur. Bu ağ input (giriş), output (çıkış) ve gizli katmanlardan oluşmaktadır. Gizli katmanlar ise 25 ve 15 düğümden oluşan iki yapıdadır. Eğitim yüzdesi %75 olarak belirlenmiştir. Validation (doğrulama) oranı %10 olarak, test oranı %15 ve epok sayısı 10000 olarak belirlenmiştir. Kullanılan eğitim algoritması ise MATLAB derin öğrenme aracı içerisinde yer alan ölçekli eşlenik gradyan geri yayılımı (trainscg) algoritmasıdır. Algoritmaya ilişkin parametre değerleri aşağıdaki gibi seçilmiştir.

```
net.trainParam.epoks=10000;
```

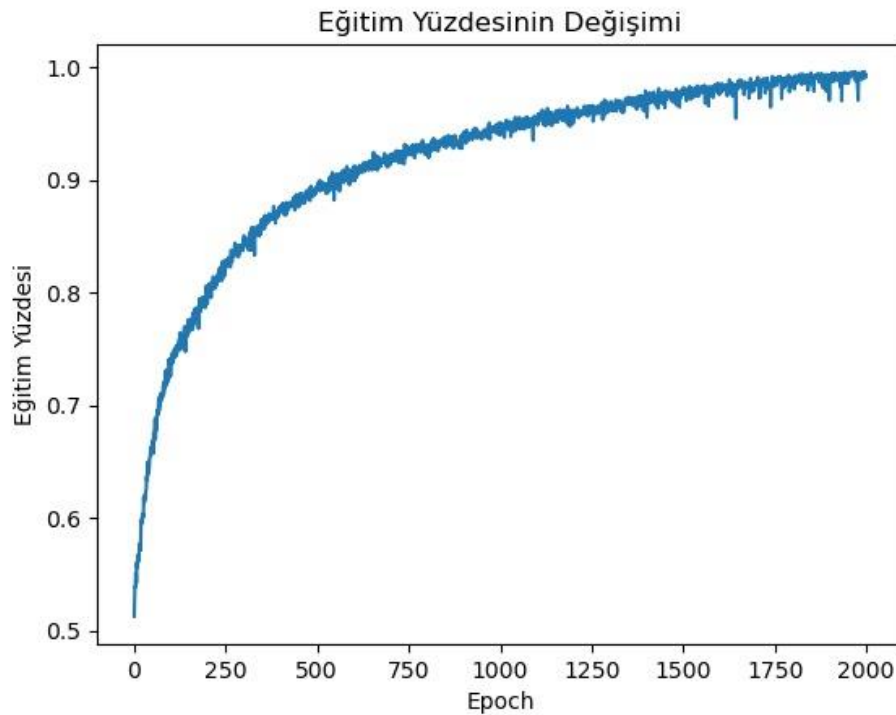
```
net.divideParam.trainRatio=75/100;
```

```
net.divideParam.valRatio=10/100;
```

```
net.divideParam.testRatio=15/100;
```

```
net.trainFcn = 'trainscg';
```

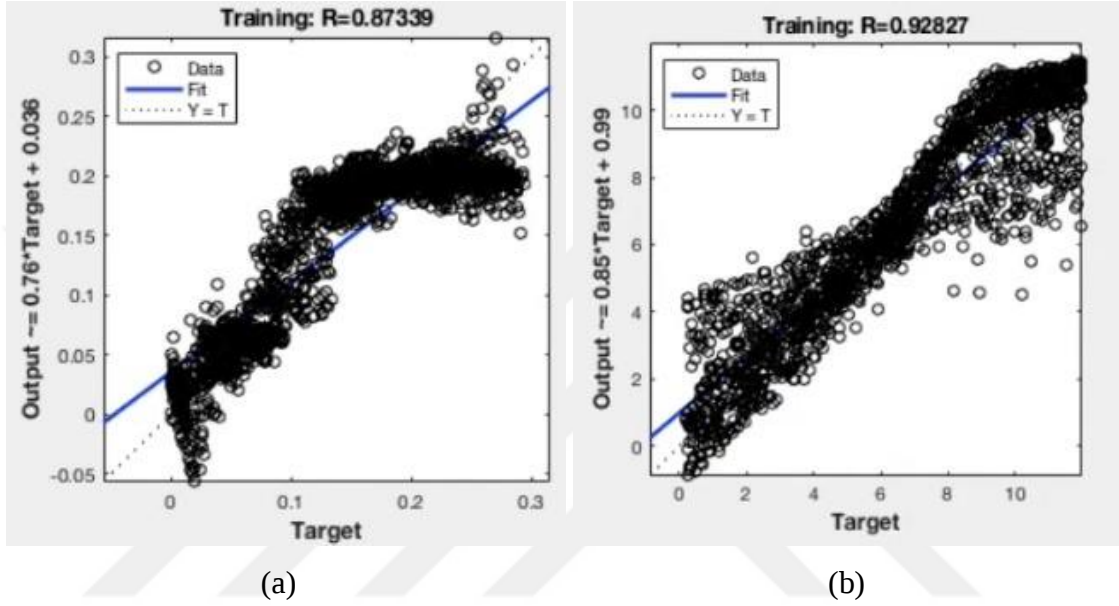
MATLAB’da oluşturulan yazılım A-Tarama ve B-Tarama için ayrı ayrı çalıştırılmış ve eğitim/test sonuçları elde edilmiştir. Eğitimde 2000 epok kullanılmış ve 0,9875 olarak elde edilen eğitim yüzdesi epok sayısına göre grafik olarak verilmiştir (Şekil 5.24).



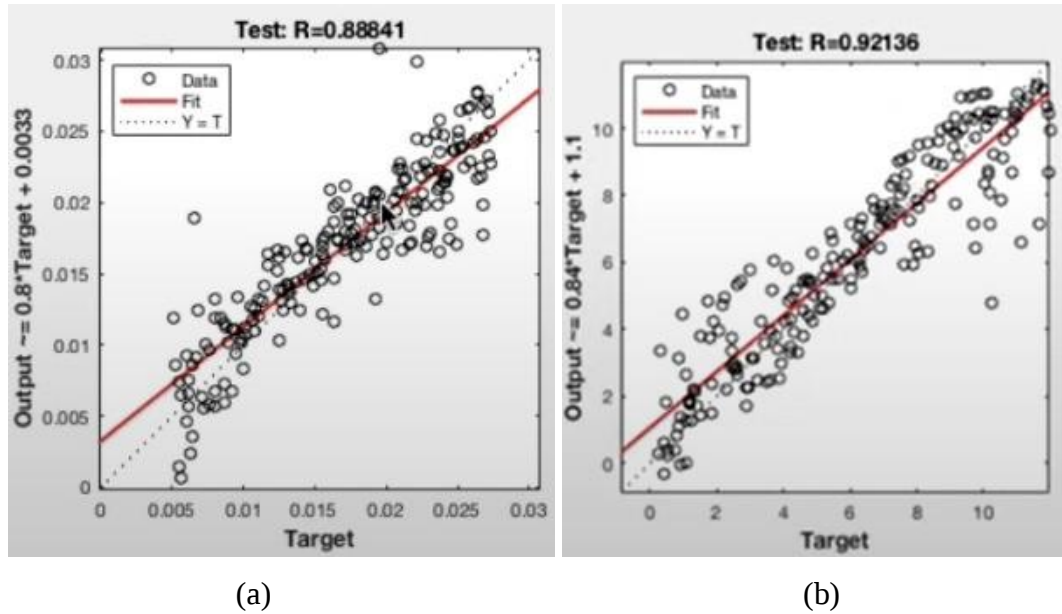
Şekil 5.24. Eğitim Yüzdesinin Epoklara Göre Değişimi

Tez çalışmasında bilgisayar tabanlı bir sistem üzerinden YNR verisinin elde etmiş olduğu sonuçların görülmesi gömülü sistem üzerinde elde edilen sonuçlar ile

karşılaştırılması açısından önemlidir. A-Tarama ve B-Tarama verisinden elde edilen eğitim ve test sonuçları aşağıdaki Şekil 5.24 ve Şekil 5.25’de gösterilmiştir. Şekillerden de görüleceği üzere B-Tarama verileri ile elde edilen eğitim (%92,82) ve test sonucu (%92,13), A-Tarama verileri ile elde edilen eğitim (%87,33) ve test sonucundan (%88,84) yüksektir. Bu durum B-Tarama tekniğinin yere nüfuz eden radar sistemlerinde daha iyi sonuç verdiğinin göstergesidir.



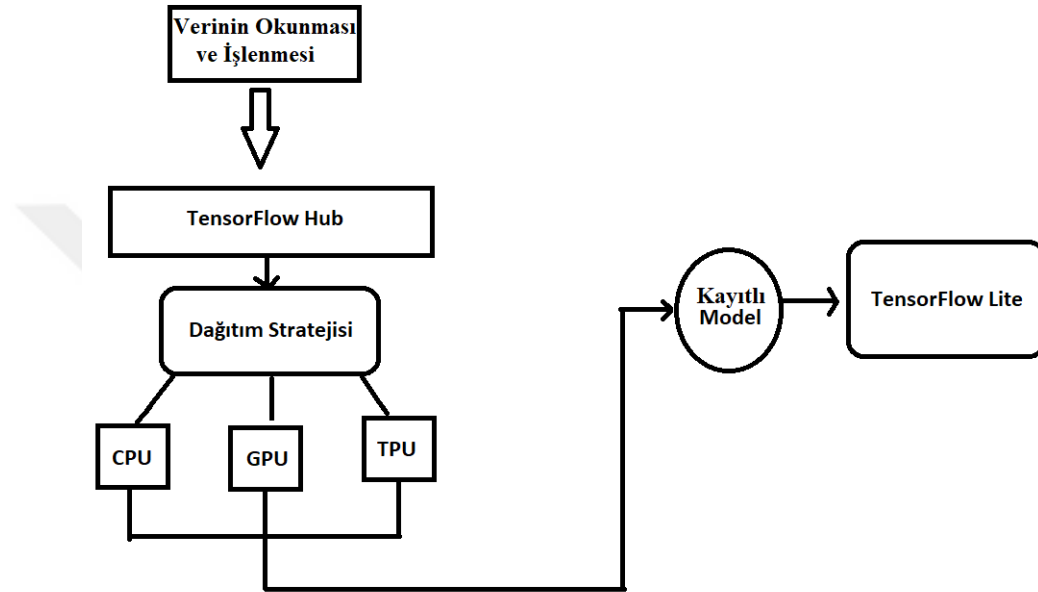
Şekil 5.24. MATLAB Üzerinde Elde Edilen Eğitim Sonucu a) A-Tarama b) B-Tarama



Şekil 5.25. MATLAB Üzerinde Elde Edilen Test Sonucu a) A-Tarama b) B-Tarama

5.7. Python ile Elde Edilen Verilerin Eğitimi

gprMax ile elde edilen verileri eğitmek için öncelikle sinir ağı yapısı kurulmuştur. Bu yapı için iki farklı ağ modeline yer verilmiştir. Bunlar, Evrişimli Sinir Ağı ve Tekrarlayan Sinir Ağıdır. Bu işlemleri yapmak için günümüzde popüler olarak kullanılan tensorflow kütüphanesi kullanılarak Google Colab platformu üzerinde gerçekleştirilmiştir. Bu senaryoyu gösteren blok şema yapısı aşağıdaki Şekil 5.26'da gösterilmiştir.



Şekil 5.26. Veri Eğitim Modelinin Genel Blok Şeması

TensorFlow (TF), Google tarafından geliştirilen açık kaynaklı bir makine öğrenimi kütüphanesidir ve sinir ağlarının oluşturulması ve eğitilmesi için kalıpları ve korelasyonları tespit etme yeteneğine sahiptir. Google Colab ise Google Research tarafından geliştirilen bir üründür ve tarayıcı aracılığıyla rastgele Python kodunun yazılmasına ve çalıştırılmasına olanak tanır. Özellikle makine öğrenimi, veri analizi ve eğitim için son derece uygundur. Google Colab, makine öğrenimi geliştiricilerine GPU ve CPU'ya ücretsiz erişim imkanı sunan bir platform olup, hem bir defter arayüzü hem de bir komut dosyası arayüzü sağlar. TensorFlow'un resmi kaynakları ayrıca Colab üzerinde çalışan birçok yardımcı materyal sunmaktadır. Colab ayrıca geliştiricilere zaman kazandırmak için çeşitli yardımcı programlar ve önceden yazılmış fonksiyonlar sağlar (Khekare ve Solanki, 2022). TF gömülü sistem kaynak kısıtlamalarının dayattığı verimlilik gereksinimlerini ve platformlar arası birlikte çalışabilirliği neredeyse imkânsız hale getiren parçalanma zorluklarını ele alır. (David ve ark., 2021). Her iki

yöntem ile de eğitim tamamlandıktan sonra gömülü sistemde kullanılmaya hazır hale getirilmiştir.

5.7.1. Eğitim Çıktısını Gömülü Sistem İçin Almak

Eğitilen model mevcut olduğu için, TF kitaplığında sağlanan TF Lite dönüştürücüyü kullanarak modelin dönüşümünü ve optimizasyonu test edilebilmektedir. Makine öğrenmesi modelinin TF'ye dönüştürülmesi için TF Lite dönüştürücü kullanıldı. Bu işlem iki adım ile gerçekleştirilmektedir. Öncelikle TF Lite dönüştürücünün nesnesini başlatılır, sonrasında başlatılan nesne bir *convert* işleviyle çağırılır. Bu işlemler aşağıdaki kod satırlarıyla tamamlanmaktadır.

#Initializing the object of the TFLite converter

TFLiteConverter = tf.lite.TFLiteConverter.from_keras_model(thesisModel)

#Calling initialized object with a convert function

tf_lite_model= TFLiteConverter.convert()

STM32 gibi işlemciler için modelin dönüşüm işlemini yaparken aşağıdaki kod satırının eklenmesinde fayda vardır. Bu kod satırı boyutu ve gecikmeyi iyileştirmek için elinden gelenin en iyisini yapacaktır. Bu seçenek girdi ve çıktıyı değil yalnızca ağırlıkları nicelleştirecektir.

Converter.optimizations = [tf.lite.Optimize.DEFAULT]

Sonrasında TF Lite, TF Lite modelini C kaynak ve başlık dosyalarına dönüştürmek için bir Python yöntemine sahiptir. Bu yöntem ile başlık ve kaynak dosyaları çıktı olarak alınacaktır. Yöntem aşağıdaki gibidir.

from tensorflow.lite.python.util import convert_bytes_to_c_source

source_text, header_text = convert_bytes_to_c_source(tf_lite_model, "thesisModel")

with open ('thesisModel.h', 'w') as file:

file.write(header_text)

with open('thesisModel.cc', 'w') as file:

file.write(source_text)

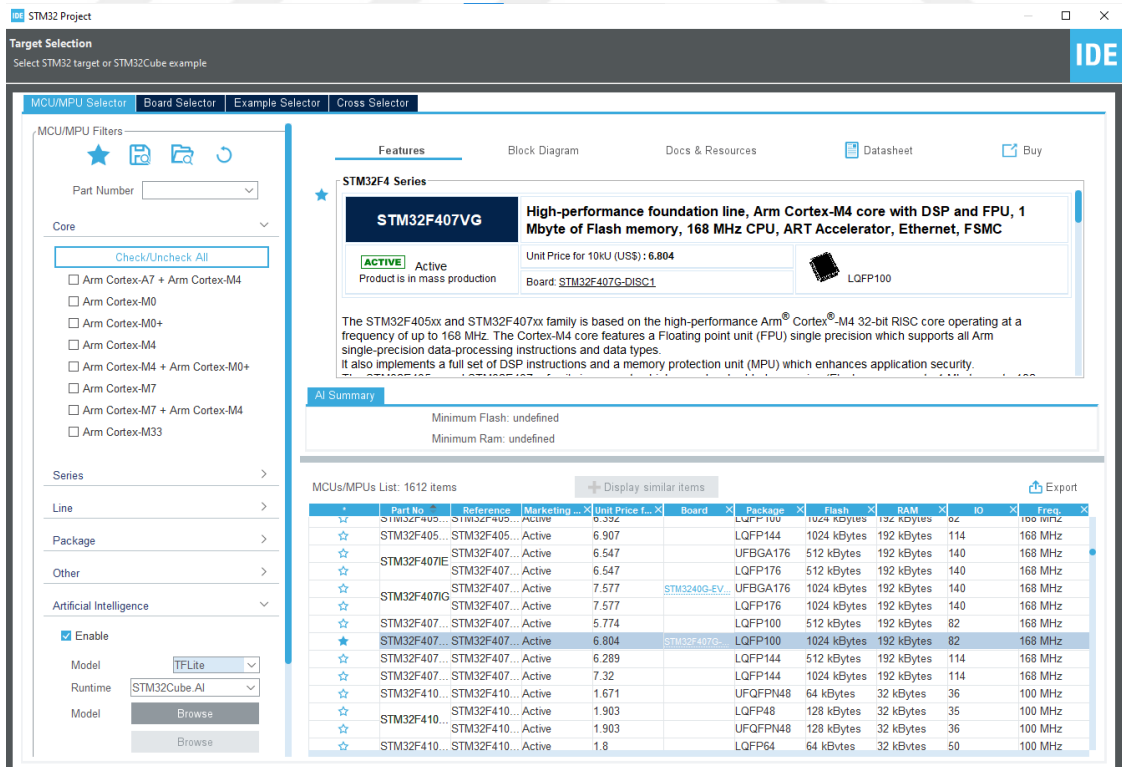
Böylelikle mikrodenetleyicide kullanılmak üzere C başlık dosyası ve kaynak dosyası oluşturulmuş olur. Sonrasında bu dosyalar indirilir ve gömülü sistem içerisinde ilgili dosya yoluna kaydedilir.

5.8. Gömülü Sistem Üzerinde Test Uygulamaları

Bu bölümde, düşük güçlü gömülü sistemlere YSA yerleştirme sorunlarını çözmek için optimizasyona odaklanılacaktır. Üç farklı işlemciler için testler yapılacak olup bu test sonuçları karşılaştırılacaktır.

5.8.1. STM32F407VGT6 ile A-Tarama Verilerinin İşlenmesi

STM32F407VGT6 işlemcisi ile A-Tarama modelinin işleme alınması ön görülmüştür. Bunun sebebi mevcut işlemcinin RAM ve ROM kapasitesinin kısıtlı olmasıdır. Ancak ST firması işlemcilerinin üzerinde yapay zeka algoritmalarının daha iyi koşturulabilmesi için X-Cube-AI eklentisini çıkarmıştır. Bu çalışmada STM32F407VGT6 işlemcisine sahip STM32F407g-disc1 Discovery geliştirme kartı kullanılmıştır. STM32CubeIDE üzerinden programlama işlemleri yeni proje ve işlemci/denetleyici seçimiyle aşağıdaki Şekil 5.28’de gösterilmiştir.



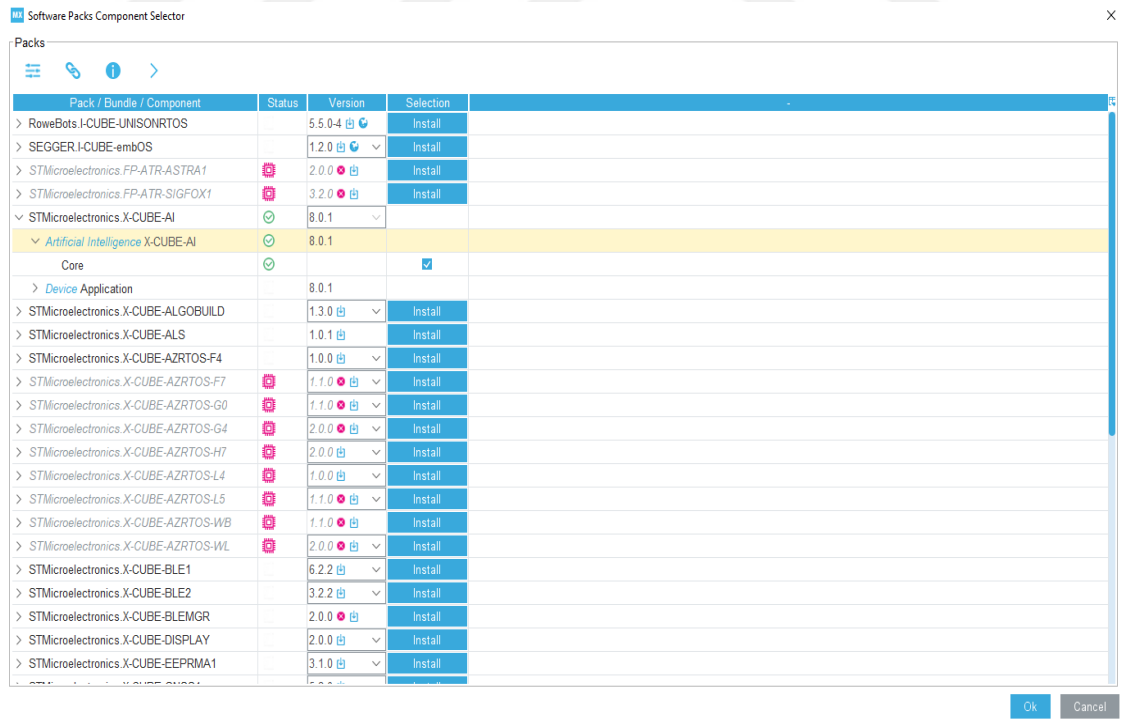
Şekil 5.28. STM32CubeIDE Üzerinden İşlemci/Denetleyici Seçimi

İşlemci ve proje seçimi yapıldıktan sonra işlemcinin config ayarlarını yapmak üzere .ioc uzantılı dosya açılmıştır. Bu dosya üzerinde işlemcinin timer ayarları yapılmıştır. Bu timerın kullanılmasındaki sebep ise mevcut sistemin performansını ölçmektir. Bu timer ayarı saat (clock) konfigürasyonuna bağlı olarak hesaplanmıştır. Bu hesabın mantığı aşağıdaki gibidir.

$$\text{Timer hızı} = \text{Bus hızı} / (\text{prescaler} + 1) \rightarrow (\text{prescaler}, \text{ön ölçekleyici})$$

$$\text{Periyot} = \text{Timer hızı} / (\text{ARR} + 1) \rightarrow (\text{ARR}, \text{Auto Reload Register}, \text{Otomatik yeniden yükleme kaydı})$$

Yine .ioc uzantılı dosya üzerinde ayarlamalar yapılırken yapay zeka (artificial intelligence) etkinleştirilmeli ve gerekli konfigürasyonlar sağlanmalıdır. Bunun için yazılım paketleri sekmesinden STMicroelectronics üreticisine ait X-Cube-AI ve versiyon seçilir (Şekil 5.29). Ardından çekirdek seçilir ve .ioc uzantılı konfigürasyon menüsünde görüntülenebilir hale gelir. Sonrasında bu menü üzerinden makine öğrenmesi için TFLite seçilir. Ardından model çağrılarak bu işlemler bitirilir.



Şekil 5.29. Stm32CubeIDE Üzerinde Yapay Zeka Eklentisinin Etkinleştirilmesi 1

Bu işlemler tamamlandıktan sonra main.c dosyası üzerinde gerekli kütüphaneler çağırılmalıdır. Bu kütüphaneler proje dosyası ile aynı dizinde olup aşağıdaki Şekil 5.30'daki gibidir.

```

/* Includes */
#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include <stdio.h>
#include "ai_datatypes.h"
#include "ai_platform.h"
#include "thesisModel.h"
#include "thesisModel_data.h"
/* USER CODE END Includes */

```

Şekil 5.30. main.c Dosyası Üzerinden Çağırılan Kütüphaneler

Kütüphaneler tanımlandıktan sonra gerekli parametrelerin ayarlanması için değişkenler tanımlanmıştır. Daha sonrasında Sinir ağı için, ara değerleri tutmak için kullanılan bellek parçası tanımlanır. Giriş ve çıkış tensörlerini depolamak için kullanılan arabellekler tanımlanmıştır. Bu aşamadan sonra çalışma belleği ayarlanır ve modelden ağırlıklar ve bias değerleri alınır. Giriş ve çıkış pointerları belirlendikten sonra uygulamaya hazır hale getirilir. Buradaki işlemleri anlatan yapı aşağıdaki Şekil 5.31'de gösterilmiştir.

```

int main(void)
{
    /* USER CODE BEGIN 1 */
    char buffer[256];
    int buffer_len[64];
    ai_error ai_err;
    ai_i32 nbatch;
    uint32_t invokeTime;
    float y_value;

    AI_ALIGNED(4) ai_u8 activationsFunction[AI_THESIS_MODEL_DATA_ACTIVATIONS_SIZE];

    AI_ALIGNED(4) ai_i8 in_ThesisData[AI_THESIS_MODEL_IN_1_SIZE_BYTES];
    AI_ALIGNED(4) ai_i8 out_ThesisData[AI_THESIS_MODEL_OUT_1_SIZE_BYTES];

    ai_handle thesis_Model = AI_HANDLE_NULL;

    ai_network_params ai_params = {
        AI_THESIS_MODEL_DATA_WEIGHTS(ai_thesis_model_data_weights_get()),
        AI_THESIS_MODEL_DATA_ACTIVATIONS(activationsFunction)
    };

    ai_input[0].n_batches = 1;
    ai_input[0].data = AI_PTR(in_data);
    ai_output[0].n_batches = 1;
    ai_output[0].data = AI_PTR(out_data);
    |

    /* USER CODE END 1 */
}

```

Şekil 5.31. Parametrelerin Ayarlanması İçin Kullanılan Değişkenlerin ve Ara Değerleri Tutmak İçin Kullanılan Bellek Parçasının Tanımlandığı Kod Bloğu

Sonrasında uygulama hem X-Cube-AI hem de TF Lite ile tamamlanmış olup, test sonuçları ve süreleri aşağıdaki Şekil 5.32 ve 5.33'deki gibidir. Bu uygulamaya dayanarak açık kaynak bir yazılım olan TF Lite, X-Cube-AI'a göre daha fazla flash kullanmakta olduğu görülmüştür. Bu sebeptendir ki sonuç çıkarım zamanı X-Cube-AI'dan daha fazladır.

```

STM32 Tensorflow Lite Thesis_Model Test
Output: 0.879325 * Duration: 268
Output: 0.878654 * Duration: 267
Output: 0.881353 * Duration: 267
Output: 0.881944 * Duration: 268
Output: 0.881941 * Duration: 268
Output: 0.879423 * Duration: 268
Output: 0.869925 * Duration: 268
Output: 0.874236 * Duration: 269
Output: 0.874456 * Duration: 268
Output: 0.874931 * Duration: 268
Output: 0.882699 * Duration: 268
Output: 0.881067 * Duration: 268
Output: 0.878996 * Duration: 268
Output: 0.878580 * Duration: 268
Output: 0.880795 * Duration: 268
Output: 0.883693 * Duration: 268
Output: 0.882638 * Duration: 268
Output: 0.869872 * Duration: 268
Output: 0.877205 * Duration: 268
Output: 0.883744 * Duration: 268

```

Şekil 5.32. TF Lite ile Yapılan Çalışmanın Sonuçları

```

STM32 X-Cube-AI Thesis_Model Test
Output: 0.879325 * Duration: 204
Output: 0.878654 * Duration: 203
Output: 0.881353 * Duration: 203
Output: 0.881944 * Duration: 204
Output: 0.881941 * Duration: 204
Output: 0.879423 * Duration: 204
Output: 0.869925 * Duration: 204
Output: 0.874236 * Duration: 205
Output: 0.874456 * Duration: 204
Output: 0.874931 * Duration: 204
Output: 0.882699 * Duration: 204
Output: 0.881067 * Duration: 204
Output: 0.878996 * Duration: 204
Output: 0.878580 * Duration: 204
Output: 0.880795 * Duration: 204
Output: 0.883693 * Duration: 204
Output: 0.882638 * Duration: 204
Output: 0.869872 * Duration: 204
Output: 0.877205 * Duration: 204
Output: 0.883744 * Duration: 204

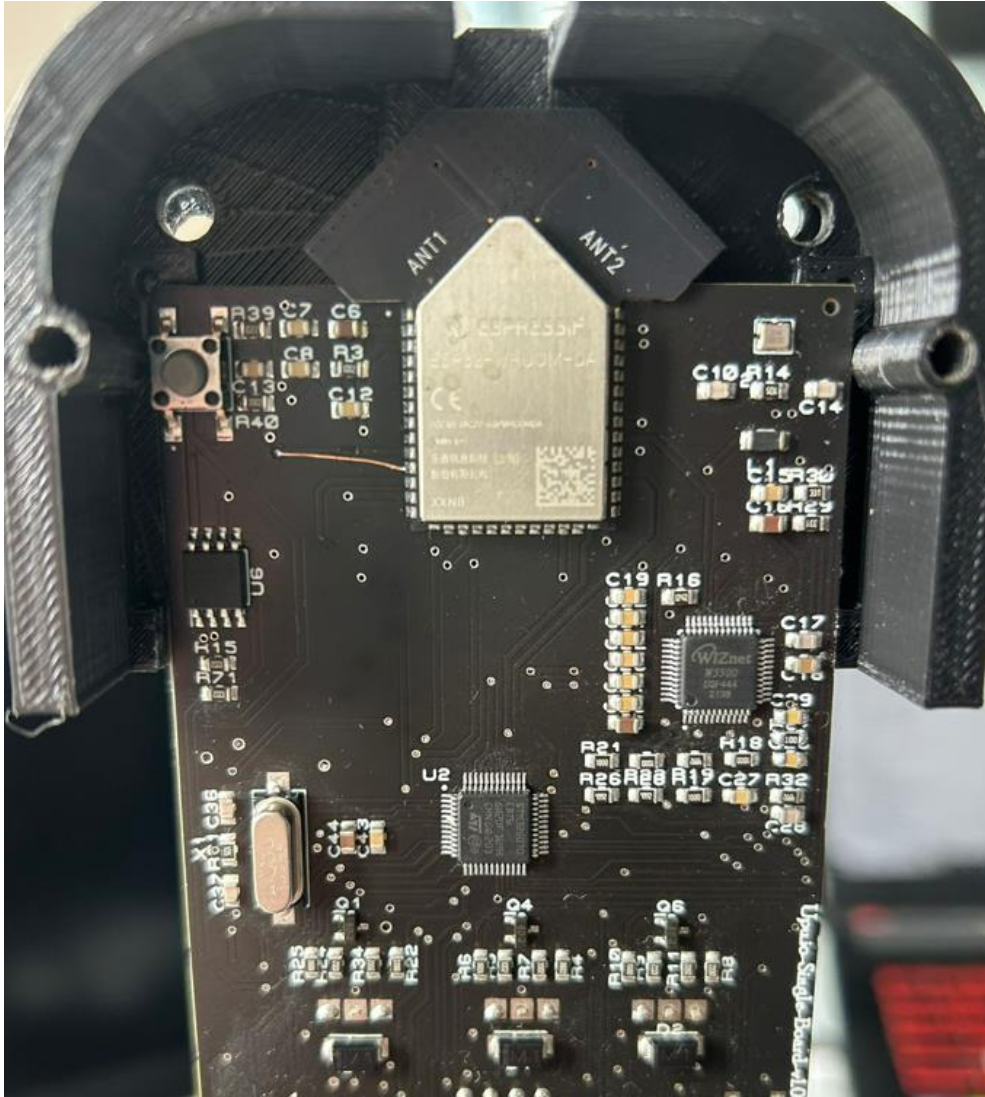
```

Şekil 5.33. X-Cube-AI ile Yapılan Çalışmanın Sonuçları

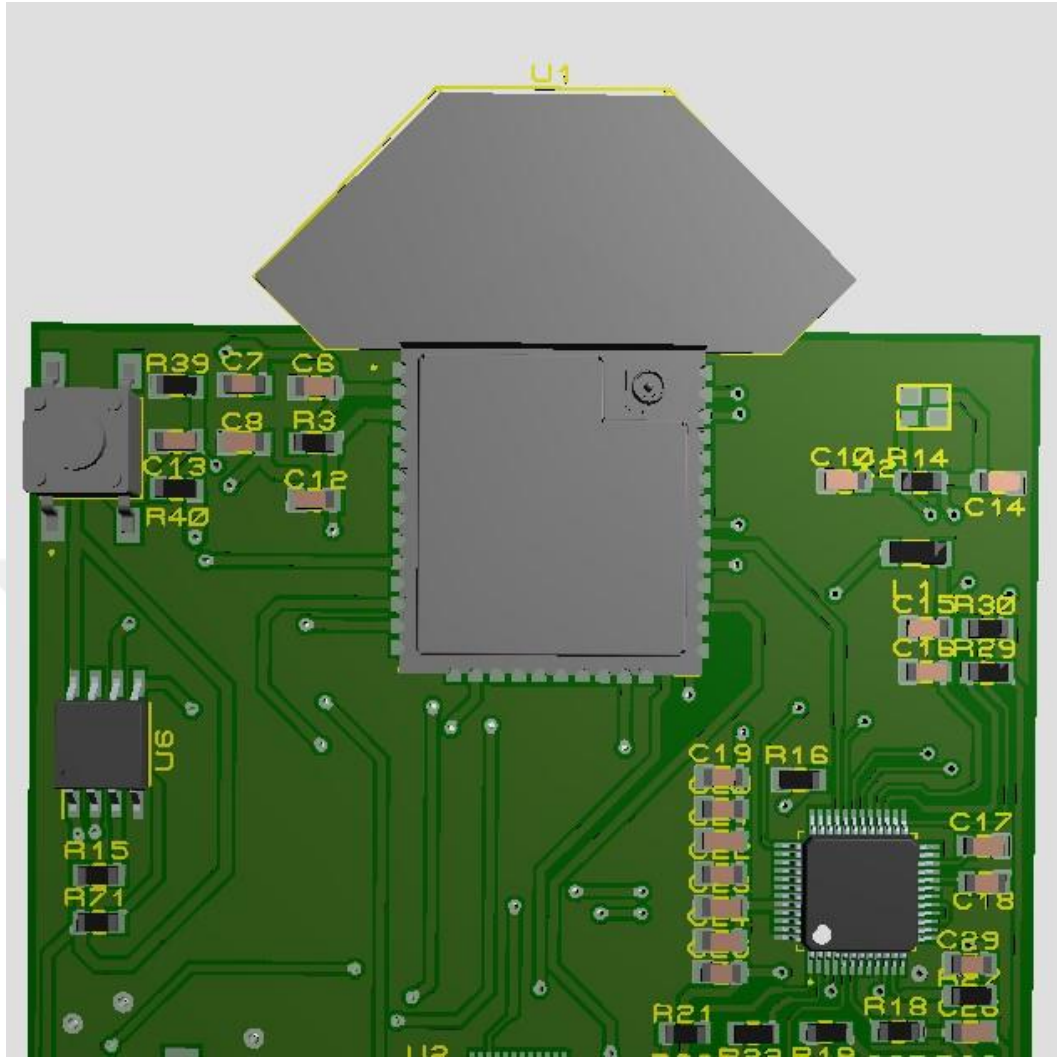
5.8.2. ESP32-WROOM-32DA ile B-Tarama Verilerinin İşlenmesi

STM32F407VGT6 işlemcisi üzerinde B-Tarama verisi ile işlem yapamama durumunun sebebi mevcut hafızanın yetersiz olması idi. Bu sebeple ile 8MB flash hafızasına sahip ESP32-WROOM-DA işlemcisi kullanarak bu işlemlerin yapılmasına

karar verilmiştir. Hafızanın zorlanması durumuna karşı external flash eklenmiştir. Ek olarak bir kullanıcı butonu konulmuş işlemcinin sıkışması gibi durumlar için kullanıcı tarafından müdahale imkânı verilmiştir. İşlemcinin sıkışması dediğimiz durum işlemci çekirdeğinin paniklemesi durumudur. Şekil 5.34 - 5.37’de ilgili çalışma kartının gerçek görseli, şematik, pcb ve üç boyutlu görseli gösterilmiştir.

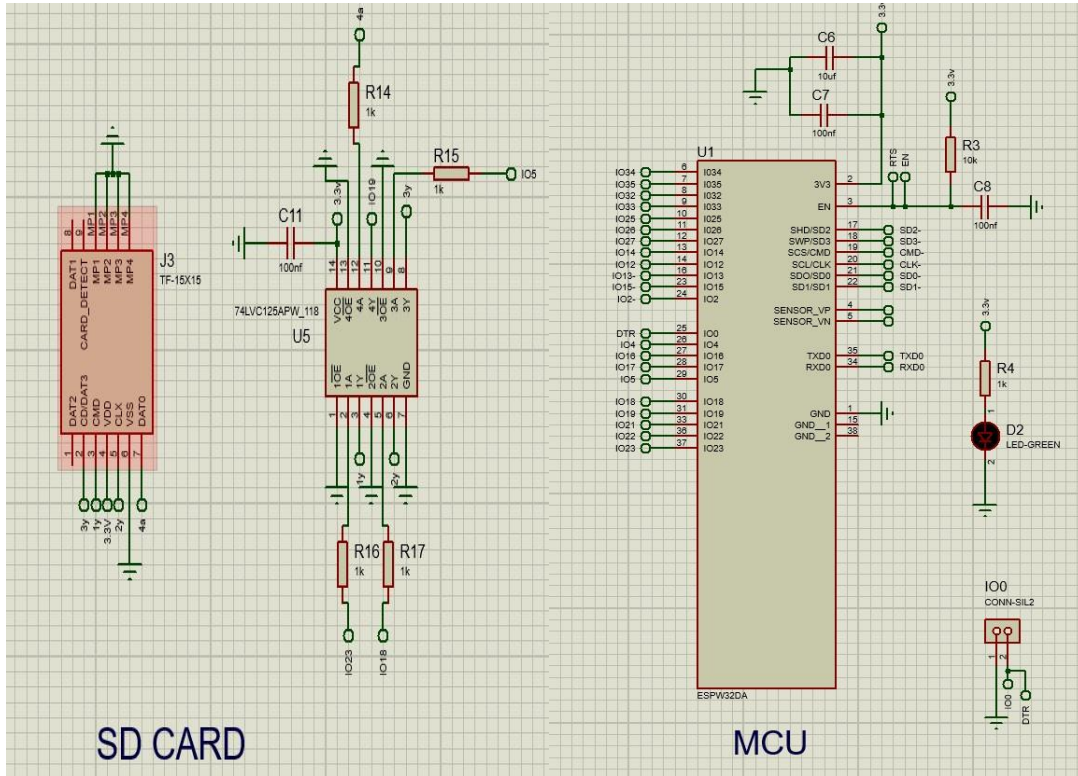


Şekil 5.34. Çalışma Kartının Gerçek Görseli

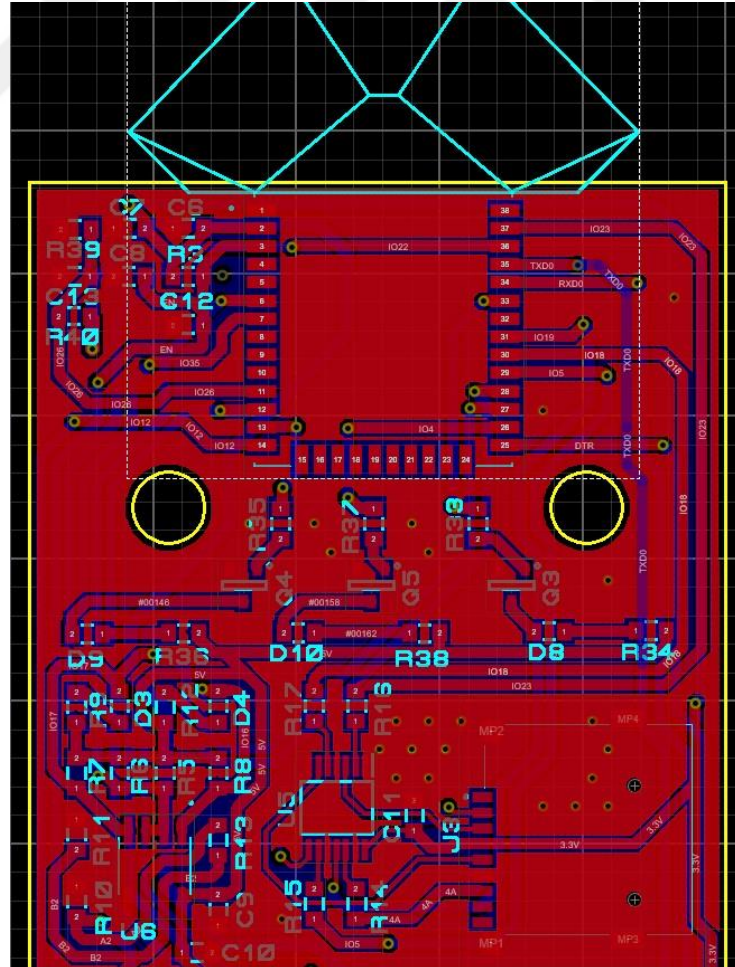


Şekil 5.37. Çalışma Kartının Üç Boyutlu Görseli

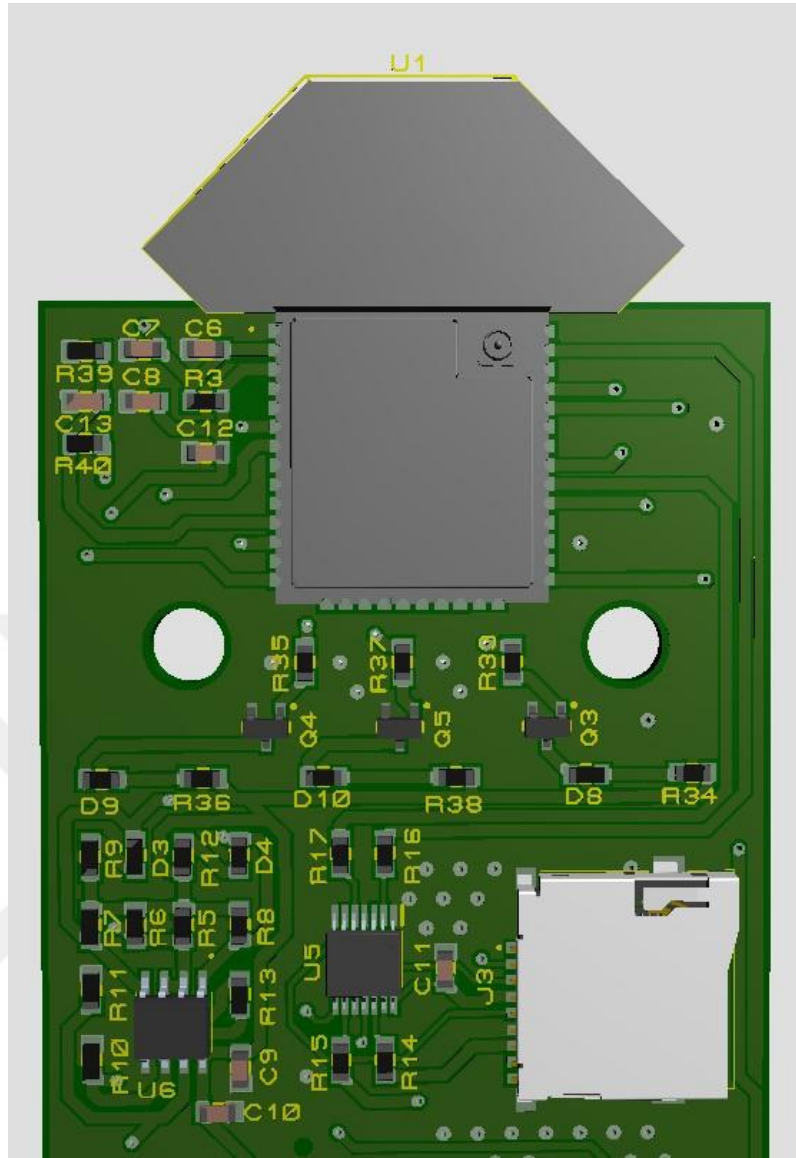
ESP32-WROOM-DA işlemcisi ram bakımından STM32F407VGT6 işlemcisinden daha üstündür. Bu sebep ile B-Tarama verileri ile elde edilen model bu işlemci üzerinde kullanılmak istenmiştir. Ancak external flash kullanılmasına rağmen taşmalar ile karşılaşılmıştır. Bu sebep ile external flash yerine sd kart kullanılması daha doğru görülmüştür. Bu uygulama için yapılan tasarım şematik, pcb ve üç boyutlu görselleri Şekil 5.38 – 5.40’da gösterilmiştir.



Şekil 5.38. SD Kart ve ESP32 Tasarımın Şematik Çizimi



Şekil 5.39. SD Kartlı Tasarımın PCB Çizimi



Şekil 5.40. SD Kartlı Tasarımın Üç Boyutlu Görüntüsü

Gerekli donanım altyapısı sağlandıktan sonra eğitilmiş model ESP32-WROOM-DA işlemcisine yüklenir. Sonrasında kod içerisinde model çıkarılır. Bu model tensörleri için bellek atanır. Giriş tensörleri ve model ekranı etkinleştirilir. Bu işlemler aşağıdaki Şekil 5.41’de yer alan kod bloğu ile sağlanmıştır.

```

static tflite::MicroErrorReporter micro_error_reporter;
error_reporter = &micro_error_reporter;

model = tflite::GetModel(mi_data);

static tflite::ops::micro::AllOpsResolver resolver;

static tflite::MicroInterpreter static_interpreter( model,
    resolver,
    tensor_arena,
    kTensorArenaSize,
    error_reporter);
interpreter = &static_interpreter;

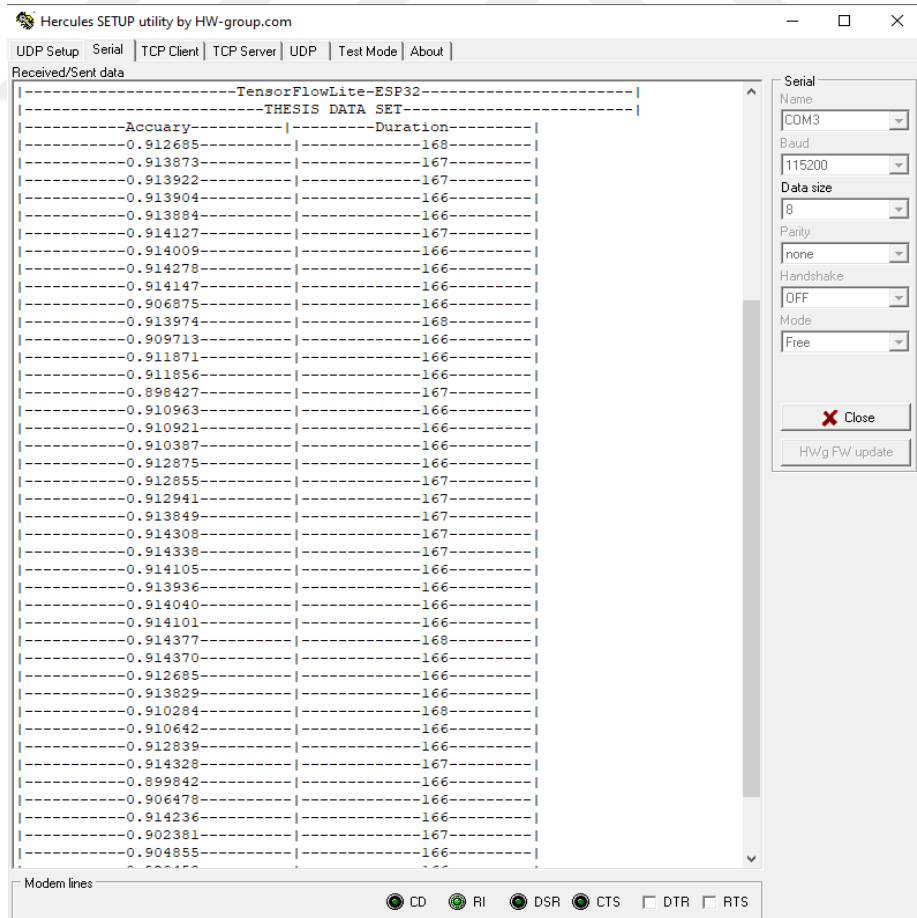
TfLiteStatus allocate_status = interpreter->AllocateTensors();

input = interpreter->input(0);
output = interpreter->output(0);

```

Şekil 5.41. Tensor Ayarları Kod Bloğu

Sonrasında uygulama TensorflowLite kütüphanesi ile tamamlanmış olup, test sonuçları ve süreleri aşağıdaki 5.42'deki gibidir.



Şekil 5.42. TensorflowLite Kütüphanesi ile Tamamlanan Uygulamanın Test Sonuçları ve Süreleri

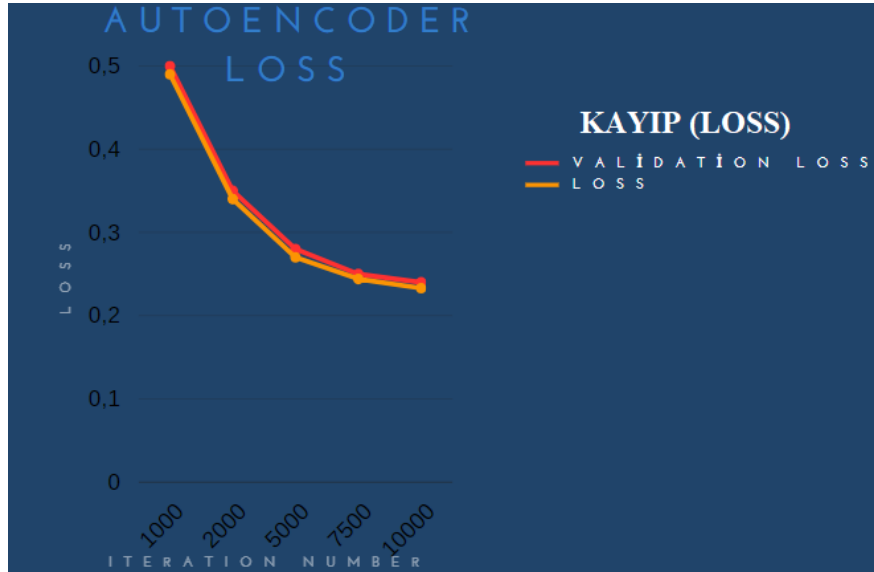
5.8.3. NVIDIA Jetson Nano ile B-Tarama Verilerinin İşlenmesi

Nvidia Jetson Nano ile B-Tarama modelinin işleme alınması ön görülmüştür. Bu çalışmada Nvidia Jetson Nano Geliştirici Kiti kullanılmıştır. Bu kite ait görsel aşağıdaki Şekil 5.43’de gösterilmiştir. Bu kit genel olarak; görüntü sınıflandırma, nesne algılama, segmentasyon ve konuşma işleme gibi uygulamalar için birden çok nöral ağı paralel olarak çalıştırmaya imkan tanıyan küçük, güçlü bir bilgisayardır. Bu bilgisayar üzerinde Colab tarafından eğitilmiş verinin test ve doğrulama işlemleri gerçekleştirilmiştir. Ayrıca bu kit ile eğitim işlemi de yapılmaktadır ancak diğer iki işlemci üzerinde eğitilmiş veri ile test yapıldığı için aynı işlemler bu kit ile de sağlanarak kıyaslanmıştır.



Şekil 5.43. Jetson Nano Geliştirme Kartı

Uygulama bu kit üzerinde jupyter notebook kullanılarak yapılmıştır. Böylece birçok ekstra imkan sağlanmıştır. Örneğin aşağıdaki Şekil 5.44’de epoklardaki değişime doğrulama kaybı grafiği çizdirilmiştir.



Şekil 5.44. Epok Değişimine Göre Doğruluk Kayıpları

Jetson Nano geliştirme kiti üzerindeki çalışma Jupyter Notebook ile yapılmış olup iki farklı model için işlem yapılmıştır. CNN modeli RNN modeline göre daha karmaşık bir model olup performans ve doğruluk bakımından daha yüksek sonuçlar vermiştir.

5.9. Uygulama Sonuçları

STM32F407VGT6, ESP32-WROOM-DA ve Jetson Nano gömülü sistem üzerinde RNN ve CNN derin öğrenme modelleri ile alınan test sonuçları Çizelge 5.1’de verilmiştir. STM32F407VGT6 gömülü donanımı B-Tarama verilerini işleyebilecek kapasitede olmadığından sadece A-Tarama verilerinin sonuçları alınmış ve çizelgeye eklenmiştir.

Çizelge 5.1. Gömülü Sistem Üzerinde Uygulama Sonuçları

TARAMA YÖNTEMLERİ				
İşlemci	RNN		CNN	
	A-Tarama	B-Tarama	A-Tarama	B-Tarama
STM32F407VGT6	0,694317	-	0,883744	-
ESP32-WROOM-DA	-	0,734864	-	0,914377
Jetson Nano	-	0,754065	-	0,925740

İşlemci	UYGULAMA SÜRELERİ (ms)	
	A-Tarama	B-Tarama
STM32F407VGT6	204	-
ESP32-WROOM-DA	-	166
Jetson Nano	-	59

Çizelgeden de görüldüğü üzere genel anlamda B-Tarama ile alınan veriler üzerinde yapılan tespit işlemi A-Tarama ile alınan verilere göre çok daha başarılı sonuç vermiştir. Yine CNN modelinden alınan sonuçlar RNN modelinden alınan sonuçlara göre daha iyi sonuç vermiştir.

Testlere göre en iyi sonuç Jetson Nano üzerinde B-Tarama verilerinin CNN modeli ile çalıştırılması sonucu elde edilmiştir (% 92,574). En kötü sonucu ise STM32F407VGT6 gömülü donanımı üzerinde A-Tarama verilerinin RNN modeli ile çalıştırılması sonucu elde edilmiştir (% 69,43). Uygulama sürelerine baktığımızda ise yine Jetson Nano gömülü sisteminin diğer (STM32 ve ESP32) gömülü kartlara göre çok daha hızlı sonuç verdiğini görmekteyiz.



6. SONUÇLAR VE ÖNERİLER

6.1 Sonuçlar

Bu çalışmada yer altında bulunan nesnelerin tespitine yönelik gömülü donanım tabanlı yere nüfuz eden radar verisi analiz sistemi geliştirilmiştir. Sistem A-Tarama ve B-Tarama tekniği ile elde edilen radar verilerini işleyerek yer altındaki nesnenin tespit edilmesini sağlamaktadır.

Uygulamanın simüle edilmesinde elektromanyetik dalga yayılımını simüle eden açık kaynak kodlu GprMax yazılımı kullanılmıştır. Veri görselleştirme amacıyla ise yine açık kaynak kodlu Paraview uygulaması tercih edilmiştir.

Çalışmanın yazılım kısmında ise makine öğrenmesi tekniklerinden Yapay Sinir Ağları ve Derin Öğrenme metotları (Evrışimli sinir ağı ve Tekrarlayan sinir ağı) kullanılarak nesnenin tespit edilmesi sağlanmıştır. Öncelikle Matlab yazılımında yer alan YSA araç kutusu kullanılarak A-Tarama ve B-Tarama verileri sınıflandırılmaya çalışılmıştır. Daha sonra Google Colab platformu üzerinde Tensorflow derin öğrenme modeli oluşturulmuş ve oluşturulan model gömülü donanımlar (STM32, ESP32 ve Jetson Nano) üzerinde eğitilerek test edilmiştir.

Çalışmada alınan sonuçlara baktığımızda genel anlamda B-Tarama ile alınan radar verilerinin A-Tarama ile alınanlara göre oldukça yüksek başarı ile nesne tespiti yaptığı görülmüştür.

Matlab platformunda YSA araç kutusu ile ileri beslemeli geri yayımlı model üzerinde yapılan uygulamada B-Tarama verileri ile elde edilen eğitim (%92,82) ve test sonucu (%92,13), A-Tarama verileri ile elde edilen eğitim (%87,33) ve test sonucundan (%88,84) yüksektir. Bu durum B-Tarama tekniğinin yere nüfuz eden radar sistemlerinde daha iyi sonuç verdiğinin göstergesidir.

Diğer bir uygulamada ise GprMax ile elde edilen veriler Evrişimli sinir ağı (CNN) ve Tekrarlayan sinir ağı (RNN) modelleri üzerinde test edilmiştir. Testler 3 farklı gömülü donanım üzerinde yapılmıştır. İlk donanım olan STM32F407VGT6 işlemcili kart B-Tarama verilerini işleyecek kapasiteye sahip olmadığından yalnızca A-Tarama verilerinde test edilmiştir. Diğer 2 gömülü donanımda ise (ESP32-WROOM-DA ve Jetson Nano) B-Tarama verileri kullanılmıştır.

Gömülü donanımlarda alınan sonuçlara baktığımızda genel anlamda B-Tarama radar verilerinin analizinden elde edilen sonuçlar A-Tarama verilerine göre çok daha yüksektir. Benzer şekilde CNN modelinden alınan sonuçlar RNN modeline göre daha

başarılıdır. Testlere göre en iyi sonuç Jetson Nano üzerinde B-Tarama verilerinin CNN modeli ile çalıştırılması sonucu elde edilmiştir (% 92,574). En kötü sonucu ise STM32F407VGT6 gömülü donanımı üzerinde A-Tarama verilerinin RNN modeli ile çalıştırılması sonucu elde edilmiştir (% 69,43). Uygulama sürelerine baktığımızda ise yine Jetson Nano gömülü sisteminin diğer (STM32 ve ESP32) gömülü kartlara göre çok daha hızlı sonuç verdiğini görmekteyiz.

6.2 Öneriler

Bu çalışmada YNR verileri (A-Tarama ve B-Tarama) üzerinde makine öğrenmesi tabanlı modeller (YSA, CNN ve RNN) uygulanmış ve gömülü sistemde test edilmiştir.

Gömülü donanımların yeterli kapasitede olmaması nedeniyle C-Tarama verileri işlenememiştir. Yeterli güce sahip gömülü donanım üzerinde C-Tarama verilerinin analizinin yapılması literatüre katkı sağlayacaktır.

Bununla birlikte farklı derin öğrenme metotlarının kullanılması da nesne tespitinde başarı oranını yükseltebilir. Ayrıca simülasyon araçları ile oluşturulan YNR verileri yerine sahada gerçek veriler üzerinde testler yapılması da literatüre önemli katkı sağlayacaktır.

KAYNAKLAR

- Anonim, 2006, Tarım istatistikleri özeti, DİE Yayınları, No;12, Ankara, 22-23.
- Griffin, S. ve Timothy, T., 2002, Ground Penetrating Radar, *Geophysical And Remote Sensing Methods For Regolith Exploration, Crc Leme Open File Report 144*, 80-89.
- Fatma Gülşah Tan, Asım Sinan Yüksel, Erdal Aydemir, Mevlüt Ersoy, European Journal of Science and Technology No. 25, pp. 159-171, August 2021
- Yenikaya, Sibel , Şölen, Ali, Bursa Uludağ Üniversitesi Bursa Uludağ Üniversitesi/Fen Bilimleri Enstitüsü/Elektrik-Elektronik Anabilim Dalı. Yere nüfuz eden radar sistemi ile gömülü hedef tespiti ve modellenmesi, <http://hdl.handle.net/11452/33111>
- Yılmaz, O. , Aydın, H. & Çetinkaya, A. (2020). Faster R-CNN Evrimsel Sinir Ağı Üzerinde Geliştirilen Modelin Derin Öğrenme Yöntemleri ile Doğruluk Tahmini ve Analizi: Nesne Tespiti Uygulaması . Avrupa Bilim ve Teknoloji Dergisi , (20) , 783-795 . DOI: 10.31590/ejosat.753896
- Ferdi Doğan İbrahim Türkoğlu, Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme Yıl 2019, Cilt: 10 Sayı: 2, 409 - 445, 20.06.2019 <https://doi.org/10.24012/dumf.411130>
- İbrahim Meşecan, Betim Çiço, İhsan Ömür Bucak, Feature vector for underground object detection using B-scan images from GprMax, Microprocessors and Microsystems, Volume 76, 2020, 103116, ISSN 0141-9331, <https://doi.org/10.1016/j.micpro.2020.103116>.
- Craig Warren, Antonios Giannopoulos, Alan Gray, Iraklis Giannakis, Alan Patterson, Laura Wetter, Andre Hamrah, A CUDA-based GPU engine for gprMax: Open source FDTD electromagnetic simulation software, Computer Physics Communications, Volume 237, 2019, Pages 208-218, ISSN 0010-4655, <https://doi.org/10.1016/j.cpc.2018.11.007>.
- Razzali, M., Joret, A., Sulong, M.S., Abdullah, M.F.L., Baharudin, E., Melor, C.K.N.A.H.C.K., Zulkefli, N.I., Shairi, N.A., 2022, Embedded iron object detection using asynchronous full wave envelope detector technique in ground penetrating radar system, *International Journal of Electrical and Computer Engineering (IJECE)*, 12 (6), 6187~6195.
- Altın, G., GPR B Tarama Görüntülerinde Gömülü Nesnelerin Bölütlenmesi ve Tespiti, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Kocaeli.
- Çolak, D., 2018, Yere Nüfuz Eden Radar Sinyallerinin İşlenmesi ile Hedef Tespiti, Başkent Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Ankara.
- Saleh, B., 2011, Introduction to Subsurface Imaging, Cambridge CB2 1TN, United Kingdom, ISBN 978-1-107-00081-0 Hardback, s.161-163.

- Daniels D. J., 2004, Ground Penetrating Radar – 2nd Edition, London, United Kingdom, s. 3.
- Anderson, D. S., Drocella, E. F., Jones, S. K., and Settle, M. A., 2001, Assessment of compatibility between ultrawideband (UWB) systems and global positioning system (GPS) receivers (report addendum). NTIA Special Publication, ISBN O 86341 360 9, 01-47.
- Lo, T., Inderwiesen, P. L., 1994, Fundamentals of Seismic Tomography, *Geophysical Monograph Series, Society of Exploration Geophysicists*, ISBN electronic: 9781560802334, 6, s. 1-44.
- Daniels D. J., 2004, Ground Penetrating Radar – 2nd Edition, London, United Kingdom, s. 3.
- Şengül, H., 2018, UHF (Aşırı Yüksek Frekans) Bandında Yere Nüfuz Eden Radar Uygulamaları İçin Geniş Bantlı Anten Tasarımı, Yüksek Lisans Tezi, Hacettepe Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü, Ankara.
- Tuncer, M.A.Ç., 2011. Basamak Frekanslı Yere İşleyen Radar (GPR) İçin Seyreklik Tabanlı Hızlı Yeraltı Görüntüleme Yöntemlerindeki Problemlere Çözümler, Yüksek Lisans Tezi, TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- Orhan, M., 2016, Çok Geniş Bantlı Radar ile Gizlenmiş Hedef Tespit ve Takibi İçin Görüntüleme Algoritmalarının Geliştirilmesi, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.
- Alper, A., Yere nüfuz eden radar (YNR) görüntülerinin elde edilmesinde zaman bölgesinde sonlu farklar metodunun kullanılması, Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Seyfried, D., Schoebel, J., 2015, Detection capability of a pulsed Ground Penetrating Radar utilizing an oscilloscope and Radargram Fusion Approach for optimal signal quality, *Journal of Applied Geophysics*, 118, s. 167-174.
- Kositsky, J., Milanfar, P., 1999, Forward-looking high-resolution GPR system, Proc. SPIE 3710, *Detection and Remediation Technologies for Mines and Minelike Targets IV*, AeroSense '99, Orlando, FL, United States.
- Kositsky, J., Cosgrove, R., Amazeen, C., Milanfar, P., 2002, Results from a forward-looking GPR mine detection system, Proceedings of SPIE - The International Society for Optical Engineering, 4742, 10.1117/12.479091.
- Ozakin, M.B., Aksoy, S., Turk, A.S., 2015, Araca monteli ileri bakan yere nüfuz eden radar sistemlerinin ZUSF yöntemi ile modellenmesi, Conference: 6. Ulusal Savunma Uygulamaları Modelleme ve Simülasyon (USMOS) Konferansı, Ankara, TÜRKİYE.

- Gurbuz A.C., Duman, M., 2015, 3D imaging for ground-penetrating radars via dictionary dimension reduction, *Turkish Journal of Electrical Engineering and Computer Sciences*, 23,(5),1242-1256.
- Ozkaya, U., Melgani, F., Bejiga, M.B., Seyfi, L., Donelli, M., 2020, GPR B scan image analysis with deep learning methods, *Measurement*, 165, 107770.
- Parrillo, R., Roberts, R.L., Haggan, A., 2005, Bridge Deck Condition Assessment Using Ground Penetrating Radar, *International Bridge Conference*, Pittsburgh, PA.
- Merkle, D., Frey, C., Reiterer, A., 2021, Fusion of ground penetrating radar and laser scanning for infrastructure mapping, *Journal of Applied Geodesy*, 5, 1, s. 31-45.
- Benedetto, A., Tosti, F., Ciampoli, L.B., D'Amico, F., 2016, An overview of ground-penetrating radar signal processing techniques for road inspections, *Journal: Signal Processing*, ISSN: 0165-1684.
- J.P. Berenger, 1994, "A Perfectly Matched Layer for the Absorption of Electromagnetic waves" *J. Comput. Phys.*, 114, 185.
- K.S. Yee, 1966, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media" *IEEE Trans. Antennas Propagat.*, AP-14, 302.
- Anonim, 2023, About the Software, gprMax, <https://www.gprmax.com/about.shtml>, Newcastle, Birleşik Krallık, [Ziyaret Tarihi: 31.05.2023].
- Kačeniauskas A., Pacevič R., Bugajev A., Katkevičius T., 2010, Efficient Visualization By Using Paraview Software On Balticgrid, *Information Technology And Control*, ISSN 1392 – 124X, 39 (2).
- Peterka T., Yu H., Ross R., et al., 2009, "ICPP - End-to-End Study of Parallel Volume Rendering on the IBM Blue Gene/P", 2009 International Conference on Parallel Processing , pg. 566, doi:10.1109/icpp.2009.27
- Moreland K., Avila L., Fisk L.A., 2007, "Parallel unstructured volume rendering in ParaView," *Proc. SPIE 6495, Visualization and Data Analysis*, 64950F (29 January 2007); <https://doi.org/10.1117/12.704533>.
- Kaisti M., Rantala V., Mujunen T., Hyrnsalmi S., Könnöla K., Makila T., Lehtonen T., 2013, Agile methods for embedded systems development - a literature review and a mapping study, *EURASIP Journal on Embedded Systems* 2013:15
- Barr, M. (1999), *Programming Embedded Systems in C and C++*, O'Reilly, Released January 1999. Publisher(s): O'Reilly Media, Inc. ISBN: 9781565923546. New York.

- STMicroelectronics. 2016. STM32F407xx Datasheet [online]. <http://www.st.com/resource/en/datasheet/stm32f407vg.pdf> [Ziyaret Tarihi: 04.04.2022].
- Espressif. 2023, ESP32-WROOM-DA datasheet [online], https://www.espressif.com/sites/default/files/documentation/esp32-wroom-da_datasheet_en.pdf, [Ziyaret Tarihi: 05.04.2022].
- Nvidia, 2022, NVIDIA Jetson Nano [online], <https://developer.nvidia.com/downloads/embedded/dlc/jetson-nano-system-module-datasheet>, [Ziyaret Tarihi: 05.05.2023].
- R. Iguernaissi, D. Merad, K. Aziz, P. Drap, People tracking in multi-camera systems: a review. *Multimed. Tools Appl.* 78(8), 10773–10793 (2019). <https://doi.org/10.1007/s11042-018-6638-5>
- H. Zhang, Z. Zhang, L. Zhang, Y. Yang, Q. Kang, D. Sun, Object tracking for a smart city using IoT and edge computing. *Sensors.* 19(9), 1987 (2019). <https://doi.org/10.3390/s19091987>
- Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* 37(9), 1834–1848 (2015). <https://doi.org/10.1109/TPAMI.2014.2388226>
- Ji, Q., Dai C., Hou C., Li X., 2021, Real-time embedded object detection and tracking system in Zynq SoC, *EURASIP Journal on Image and Video Processing* 21 (2021). <https://doi.org/10.1186/s13640-021-00561-7>
- Kalmet, PHS., Sanduleanu, S., Primakov, S., Wu, G., Jochems, A., Reafae T., Ibrahim A., Hulst Lv., Lambin, P., Poeze M., 2020, Deep learning in fracture detection: a narrative review, *Acta Orthopaedica*, 91 (2020) – Issue 2, 215-220.
- Barla N., 2021, *A Gentle Introduction to Deep Learning—the ELI5 Way* [online], v7labs, Bangalore, Hindistan, <https://www.v7labs.com/blog/deep-learning-guide>, [Ziyaret Tarihi: 07.05.2023].
- Şengöz N., 2017, Yapay Sinir Ağları, <https://www.derinogrenme.com/2017/03/04/yapay-sinir-aglari/>, Türkiye, [Ziyaret Tarihi : 14.05.2023].
- Kılıç H., 2021, Yapay Sinir Ağları ile Örüntü Tanıma, <https://www.hayrikilic.net/2021/04/27/yapay-sinir-aglari-ile-oruntu-tanima/>, Ankara, Türkiye, [Ziyaret Tarihi: 31.05.2023].
- Ozturk, V., 2017, *Yapay Sinir ağlarının Mimarisi ve Yapı Elemanları* [online], Yapay Zeka, Turkey, <http://volkanozturk.net/yapay-sinir-aglarinin-mimarisi-ve-yapi-elemanlari/> [Ziyaret Tarihi: 31 Mayıs 2023].

- Sharma, V., Rai, S., Dev, A., 2012, A Comprehensive Study of Artificial Neural Networks, *International Journal of Advanced Research in Computer Science and Software Engineering*, 2, 10, ISSN: 2277 128X.
- Yıldırım E., 2020, *Yapay Sinir Ağı(Artificial Neural Network) Nedir?* [online], Veri Bilimi Okulu, <https://www.veribilimiokulu.com/blog/yapay-sinir-agiartificial-neural-network-nedir/>, Türkiye, [Ziyatet Tarihi: 1 Haziran 2023].
- Okumuş R., 2019, Convolutional Neural Networks (Evrişimsel Sinir Ağları) [online], Medium, <https://medium.com/@rabiaokumus96/convolutional-neural-networks-evri%C5%9Fimsel-sinir-a%C4%9Flar%C4%B1-cceb887a2979>, [Ziyatet Tarihi: 2 Haziran 2023].
- Sezgin N., Tekin R., Çalışkan A., 2012, Örüntü Tanımada Hopfield Ağı'nın Kullanılması, *Batman University Journal of Life Sciences*, 1 (2) , 257-266.
- Karakuş B.A., 2018, Derin Öğrenme Ve Büyük Veri Yaklaşımları İle Metin Analizi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Elazığ, Türkiye.
- Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2), 234-242.
- Şeker, A., Diri, B., & Balık, H. H. (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.
- Doğan F., ve Türkoğlu İ., 2019, Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme, *DÜMF Mühendislik Dergisi* 10:2 (2019) : 409-445, doi: 10.24012/dumf.411130.
- Siwek, T., 2019, gprMax-Designer [online], <https://github.com/tomsiwek/gprMax-Designer>, [Ziyatet Tarihi: 2 Haziran 2023].
- Savita S.J., Pallavi A., 2022, "Modeling of GPR Using gprMax Simulation," 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, pp. 1-4, doi: 10.1109/ICDCECE53908.2022.9792883.
- Giannakis I., Warren C., Giannopoulos A., 2021, An open source electromagnetic simulator for generating big data for Ground Penetrating Radar applications [online], https://presentations.copernicus.org/EGU21/EGU21-10347_presentation.pdf, [Ziyatet Tarihi: 2 Haziran 2023].
- Giannakis I., 2020, Sentetik verileri kullanarak makine öğrenimi [online], University of Aberdeen, Online Oturum, UK.
- Khekare G., Solanki K., 2022, Real Time Object Detection With Speech Recognition Using Tensorflow Lite, *Gis science journal*, 9 (3), 552-559, doi:20.18001.GSJ.2022.V9I3.22.38824.

- David R., Duke J., Jain A., Reddi V.J., Jeffries N., Li J., Kreeger N., Nappier I., Natraj M., Wang T., Warden P., Rhodes R., 2021, TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems, Proceedings of Machine Learning and Systems, 3 (MLSys 2021).
- Anonymous, 1989, Farm accountancy data network, an A-Z of methodology” Commission Report of the EC, Brussels, 16-19.
- Corliss, R., 1993, *Pacific Overtures Times*, 142 (11), 68-70.
- Dasgupta, D., 1998, Artificial immune systems and their applications, *Springer-Verlag*, Berlin - Heidelberg, 45-52.
- De Castro, L. N. and Von Zuben, F. J., 2000, Artificial immune systems: Part I- Basic theory and applications, *DCA-RT 02/00, Brasil*, 23-28.
- Güneş, S. ve Polat, K., 2009, Elektrokardiyogram (EKG) aritmi teşhisinde en az kareli destek vektör makinaları kullanımına dayalı medikal teşhis destek sistemi, *13. Biyomedikal Mühendisliği Ulusal Toplantısı, BİYOMUT-2009*, İstanbul, 170-173.
- Holland, M., 2002, *Guide to citing Internet sources* [online], Poole, Bournemouth University, http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sourc.html [Ziyaret Tarihi: 4 Kasım 2002].
- Mason, J., 1832, Map of the countries lying between Spain and India, 1:8.000.000, London: Ordnance Survey.
- Özgören, M., 2006, Flow Structure in the downstream of square and circular cylinders, *Flow Measurement and Instrumentation*, 17 (4), 225-235.
- European Journal of Science and Technology No. 25, pp. 159-171, August 2021
Copyright © 2021 EJOSAT Fatma Gülşah Tan, Asım Sinan Yüksel, Erdal Aydemir, Mevlüt Ersoy
- Bursa Uludağ Üniversitesi Yenikaya, Sibel , Şölen, Ali, Bursa Uludağ Üniversitesi/Fen Bilimleri Enstitüsü/Elektrik-Elektronik Anabilim Dalı. Yere nüfuz eden radar sistemi ile gömülü hedef tespiti ve modellenmesi, <http://hdl.handle.net/11452/33111>
- Yılmaz, O. , Aydın, H. & Çetinkaya, A. (2020). Faster R-CNN Evrimsel Sinir Ağı Üzerinde Geliştirilen Modelin Derin Öğrenme Yöntemleri ile Doğruluk Tahmini ve Analizi: Nesne Tespiti Uygulaması . Avrupa Bilim ve Teknoloji Dergisi , (20) , 783-795 . DOI: 10.31590/ejosat.753896
- Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme Yıl 2019, Cilt: 10 Sayı: 2, 409 - 445, 20.06.2019 <https://doi.org/10.24012/dumf.411130>