

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**RADAR MİKRO-DOPPLER İMZA SINIFLANDIRMA PROBLEMİ İÇİN
DERİN SİNİR AĞI BAŞLATMA VE EĞİTİM METODOLOJİLERİ**

YÜKSEK LİSANS TEZİ

Mehmet Saygın SEYFİOĞLU

Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Ayşe Melda YÜKSEL TURGUT

ARALIK 2017

Fen Bilimleri Enstitüsü Onayı

.....
Prof. Dr. Osman EROĞUL
Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

.....
Doç. Dr. Tolga GİRİCİ
Anabilimdalı Başkanı

TOBB ETÜ, Fen Bilimleri Enstitüsü'nün 141211047 numaralı Yüksek Lisans öğrencisi **Mehmet Saygın SEYFİOĞLU**'nın ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"RADAR MİKRO-DOPPLER İMZA SIFLANDIRMA PROBLEMİ İÇİN DERİN SINIR AĞI BAŞLATMA VE EĞİTİM METODOLOJİLERİ"** başlıklı tezi 06.12.2017 tarihinde aşağıda imzaları olan jüri tarafından kabul edilmiştir.

Tez Danışmanı: **Doç. Dr. Ayşe Melda YÜKSEL TURGUT**
TOBB Ekonomi ve Teknoloji Üniversitesi

Eş Danışman: **Yrd. Doç. Dr. Ahmet Murat ÖZBAYOĞLU**.....
TOBB Ekonomi ve Teknoloji Üniversitesi

Jüri Üyeleri: **Prof. Dr. Çağatay CANDAN (Başkan)**
Orta Doğu Teknik Üniversitesi

Prof. Dr. Bülent TAVLI
TOBB Ekonomi ve Teknoloji Üniversitesi

Yrd. Doç. Dr. Mehmet TAN
TOBB Ekonomi ve Teknoloji Üniversitesi

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, alıntı yapılan kaynaklara eksiksiz atıf yapıldığını, referansların tam olarak belirtildiğini ve ayrıca bu tezin TOBB ETÜ Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Mehmet Saygın SEYFİOĞLU

ÖZET

Yüksek Lisans Tezi

RADAR MİKRO-DOPPLER İMZA SINIFLANDIRMA PROBLEMİ İÇİN DERİN SİNİR AĞI BAŞLATMA VE EĞİTİM METODOLOJİLERİ

Mehmet Saygın SEYFİOĞLU

TOBB Ekonomi ve Teknoloji Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Ayşe Melda YÜKSEL TURGUT

Tarih: ARALIK 2017

Son zamanlarda, derin yapay sinir ağları (DNN), sentetik açıklılı radar (SAR) veya mikro-Doppler tabanlı otomatik hedef tanıma gibi RF sinyal sınıflandırma uygulamalarında araştırma konusu olmuştur. Bununla birlikte, radar verisi elde etmenin masraflı ve zor olması, elde edilebilecek veri sayısını kısıtlamaktadır. DNN'ler yüksek varyanslı modeller olmaları sebebiyle aşırı öğrenmeye yatkınlara. Dolayısıyla iyi bir genelleme öğrenmek için fazla miktarda etiketlenmiş veriye ihtiyaç duymaktadırlar. Bu tez çalışmasında, ilk olarak 4 GHz sürekli dalga radarı 12 farklı ancak birbirine benzer insan hareketi için bir veri seti oluşturulmuştur. Sonrasında literatürde az sayıda veriyle DNN eğitmek için önerilen iki temel teknik kıyaslanmıştır: Gözetimsiz ön-eğitim ve aktarımlı öğrenme. Gözetimsiz ön-eğitim bir konvolüsyonel otokodlayıcı (CAE) yapısı üzerinden, aktarımlı öğrenme ise ImageNet üzerine eğitilmiş popüler CNN mimarileri (VGGNet, GoogleNet and ResNet) üzerinden gerçekleştirilmiştir. Buna ek olarak, aktarımlı öğrenmenin benzer transfer alanı üzerinden uygulanabilmesi için, farklılaştırılmış radar mikro-Doppler simülasyonları oluşturan yenilikçi bir metot önerilmiştir. Tüm sonuçlar, el işçiliği ile oluşturulmuş özniteliklerin çıkartılarak bir SVM sınıflandırıcısının eğitildiği geleneksel yöntem ile kıyaslanmıştır. Tezin sonunda, önerilen artçıl aktarımlı öğrenme ağı DivNet'in, diğer DNN mimarileri arasında en yüksek sınıflandırma doğruluğunu sağladığı ve gürültüye karşı gürbüz olduğu gösterilmiştir.

Anahtar Kelimeler: Aktarımlı öğrenme, Derin sinir ağları, Konvolüsyonel sinir ağları, Radar hareket sınıflandırma, Mikro-Doppler, Oto-kodlayıcı

ABSTRACT

Master of Science

DEEP NEURAL NETWORK INITIALIZATION AND TRAINING METHODOLOGIES FOR RADAR MICRO DOPPLER SIGNATURE CLASSIFICATION

Mehmet Saygın SEYFİOĞLU

TOBB University of Economics and Technology
Institute of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

Supervisor: Assoc. Prof. Ayşe Melda YÜKSEL TURGUT

Date: December 2017

Recently, deep neural networks (DNNs) have been the subject of intense research for the classification of radio frequency (RF) signals, such as synthetic aperture radar (SAR) imagery or micro-Doppler based automatic target recognition. However, a fundamental challenge is the typically small amount of data available due to the high costs and resources required for measurements. Deep neural networks (DNNs), however, require large-scale labeled datasets to prevent overfitting while having good generalization. In this thesis, the efficacy of two neural network initialization techniques - unsupervised pre-training and transfer learning - for dealing with training DNNs on small data sets are compared. Unsupervised pre-training is implemented through the design of a convolutional autoencoder (CAE), while transfer learning from two popular CNN architectures (VGGNet, GoogleNet and ResNet). Also, a novel method for generating diversified radar micro-Doppler signatures using Kinect-based motion capture simulations is proposed as a training database for DNNs exploiting transfer learning. Results have also been compared with the conventional approach where handcrafted features are employed to train an SVM classifier. In particular, it is shown that the proposed residual transfer learning network, DivNet, offers the highest classification accuracy among other DNN architectures and, is robust to image noise.

Keywords: Transfer learning, Deep neural networks, Convolutional neural networks, Radar gait classification, Micro-Doppler, Autoencoder

TEŐEKKÜR

Her Őeyden önce bu günlere gelmemi sađlayan, desteklerini bir an olsun esirgemeyen, ne olursa olsun yanımda olan kıymetli annem ve babama sevgi ve saygılarımı sunarım.

Üniversite eđitimim ve araŐtırmalarım boyunca desteklerini ve fikirlerini bir an olsun esirgemeyen, her konuŐmamızda ufkumu ačan deđerli hocalarım Yrd. Doç. Sevgi Zübeyde GÜRBÜZ, Doç. Dr. Ali Cafer GÜRBÜZ, Yrd. Doç. Dr. Ahmet Murat ÖZBAYOĐLU ve Doç. Dr. AyŐe Melda Yüksel TURGUT 'a sonsuz teŐekkürlerimi sunarım. Ayrıca sunduđu güzel çalıŐma ortamı ve burs imkanı ile beni destekleyen deđerli TOBB Ekonomi ve Teknoloji Üniversitesi ailesine minnetlerimi sunarım.

Bu zorlu araŐtırma sürecinde, bana tecrübelerini aktarmaktan çekinmeyen ve her zaman yol gösteren deđerli lab ve ofis arkadaşlarım BarıŐ Erol, İlhan İhsan, Bürkan Tekeli, Batuhan Bardak, Acar Erdinç ve Semih Yađcıođlu'na teŐekkür ederim.

Hayatı eđlenceli kılan ve birlikte vakit geçirmekten çok keyif aldıđım deđerli dostlarım Sinan, İbrahim, Bülent, Özde, Seda, Abdullah, Çađrı, Onur, Batuhan, Umut ve Acar'a çok teŐekkürler.

İÇİNDEKİLER

| | <u>Sayfa</u> |
|--|--------------|
| ÖZET | iv |
| ABSTRACT | iv |
| TEŞEKKÜR | vi |
| İÇİNDEKİLER | vii |
| ŞEKİL LİSTESİ | ix |
| ÇİZELGE LİSTESİ | x |
| KISALTMALAR | xi |
| SEMBOL LİSTESİ | xii |
| 1. GİRİŞ | 1 |
| 1.1 Problem Tanımı ve Motivasyon | 1 |
| 1.2 Mikro-Doppler Etkisi | 2 |
| 1.3 Geleneksel Yöntem | 2 |
| 1.4 Güncel Literatür ve Önerilen Yöntem | 3 |
| 1.5 Tez Kapsamı | 5 |
| 2. TEZ KAPSAMINDA OLUŞTURULMUŞ VERİ SETLERİ | 7 |
| 2.1 Deneysel Veri Seti | 7 |
| 2.2 Aktarımlı Öğrenme İçin Simülasyon Veri Setinin Oluşturması | 8 |
| 2.2.1 Kinect tabanlı simülasyon veri seti | 9 |
| 2.2.2 Simüle Mikro-Doppler imzaları için farklılaştırma metodolojisi | 10 |
| 2.2.2.1 Boy ve hız modifikasyonları | 11 |
| 2.2.2.2 Eklem parametrelerinin belirlenmesi | 12 |
| 2.2.3 Önerilen metodolojinin validasyonu | 12 |
| 2.2.4 Oluşturulan simülasyon Mikro-Doppler veri seti | 13 |
| 2.3 Mikro-Doppler İmzaları | 14 |
| 3. GELENEKSEL YÖNTEM | 17 |
| 3.1 Öznitelik Tanımları | 17 |
| 3.1.1 Fiziksel öznitelikler | 17 |
| 3.1.2 Dönüşüm tabanlı öznitelikler | 18 |
| 3.1.3 Ses işleme tabanlı öznitelikler | 18 |
| 3.2 Öznitelik Seçimi ve Sınıflandırma Kıyaslaması | 19 |
| 4. YAPAY SİNİR AĞLARI | 21 |
| 4.1 Yapay Sinir Ağlarına Giriş | 21 |
| 4.2 Geri Yayılım Algoritması | 23 |
| 4.3 Derin Öğrenme | 25 |
| 4.4 Derin Öğrenme Mimarileri | 25 |
| 4.4.1 Otokodlayıcı | 26 |
| 4.4.2 Konvolüsyonel Yapay Sinir Ağları | 28 |
| 4.4.3 Konvolüsyonel Otokodlayıcı | 29 |
| 4.4.4 Aktarımlı Öğrenme | 31 |
| 4.5 Mikro-Doppler Simülasyon Verisi Üzerinden Aktarımlı Öğrenme | 33 |

| | |
|---|----|
| 5. DENEYSEL SONUÇLAR | 35 |
| 5.1 Geleneksel Yaklaşım Karşı Derin Öğrenme | 35 |
| 5.1.1 Sınıflandırma sonuçlarının tartışılması | 36 |
| 5.1.2 Darboğaz özniteliklerinin performansı | 37 |
| 5.2 Ağırlık Başlatma Yöntemlerinin Kıyaslanması | 37 |
| 5.2.1 Aktivasyon Haritalarının incelenmesi | 39 |
| 5.3 Önerilen Yöntemle Aktarımlı Öğrenmenin Kıyaslanması | 40 |
| 5.3.1 Darboğaz Özniteliklerin kıyaslanması | 40 |
| 5.3.2 Önerilen yöntemle aktarımlı öğrenmenin gürültü altında kıyaslanması | 41 |
| 5.4 İşlem Karmaşıklığı | 43 |
| 6. SONUÇLAR | 47 |
| KAYNAKLAR | 48 |
| ÖZGEÇMİŞ | 48 |



ŞEKİL LİSTESİ

| | <u>Sayfa</u> |
|---|--------------|
| Şekil 2.1: Radar donanımının konfigürasyonu. | 7 |
| Şekil 2.2: İncelenen insan hareketleri için spektrogram örnekleri. | 9 |
| Şekil 2.3: 15 GHz CW radar için üretilmiş Kinect Tabanlı Mikro-Doppler imzaları. | 10 |
| Şekil 2.4: Kinect verisi için pertürbasyon algoritması. | 13 |
| Şekil 2.5: Tek bir veriden çıkartılmış Kinect tabanlı simülasyon sonuçları: (a) Kısa bir kişi (b) Uzun bir kişi (c) adım hızı yavaş olan bir kişi, (d) adım hızı hızlı olan bir kişi ve bunlara sırasıyla karşılık gelen mikro-Doppler imzaları (e) Kısa (f) Uzun (g) Yavaş, ve (h) Hızlı. | 14 |
| Şekil 2.6: Önerilen imza çeşitlendirme tekniğinin doğrulanması: (a) Sınıflararası farkların kıyaslanması 1-9 Yürüme, 9-18 Koşma, 18-27 Topallama, 27-36 Düşme, 36-45 Oturma, 45-51 Bastonla yürüme ve 51-55 Değnekle yürüme) ve (b) Sınıf içi benzerliklerin kıyaslanması (x eksenini boy iterasyonlarını içermektedir: 1-114 ilk, 114-228 ikinci, 228-342 üçüncü, 342-456 dördüncü, and 456-570 beşinci)). | 14 |
| Şekil 3.1: SVM, RF ve Xgboost yöntemleri için öznelik sayısına bağlı sınıflandırma sonuçları. | 19 |
| Şekil 4.1: Mcculloch Pitts Nöronu'nun genel yapısı. | 21 |
| Şekil 4.2: Sigmoid ve Hiperbolik Tanjant aktivasyonları. | 22 |
| Şekil 4.3: Bir ileri beslemeli tam bağlı yapay sinir ağı. | 23 |
| Şekil 4.4: 3 Katmanlı AE yapısı, kodlayıcı katmanlarında sırasıyla 200-100-50 nöron, kod çözücü ise 50-100-200 nöron içermektedir. | 26 |
| Şekil 4.5: CNN ve CAE modelleri için önerilmiş filtre birleştirme tekniği. | 29 |
| Şekil 4.6: Tez kapsamında önerilen CNN mimarisi her katmanda 30 3x3'lük filtre içeren 3 katmandan meydana gelmektedir. Sonunda da 2 adet 150 nöronluk tam bağlı katman içermektedir. | 30 |
| Şekil 4.7: Tez kapsamında önerilen CAE mimarisi. Gözetimsiz ön-egitimden sonra kod çözücü çıkartılır ve kodlayıcının sonuna 2 adet tam bağlı katman ile softmax sınıflandırıcısı eklenir. | 31 |
| Şekil 4.8: GoogleNet mimarisinin yapı taşı olan inception bloğunun gösterimi [?]. | 32 |
| Şekil 4.9: Artçıl Ünitelerin yapısı (BN yığıt normalizasyonunun kısaltmasıdır). | 33 |
| Şekil 4.10: Önerilen DivNet mimarisi. | 34 |
| Şekil 5.1: AE, CNN ve CAE için validasyon performans kıyaslaması. | 35 |
| Şekil 5.2: Ağırlık başlatma tekniklerinin performansı. | 41 |
| Şekil 5.3: Ağırlık başlatma tekniklerinin eğitimde kullanılan örnek sayısına göre performansı. | 41 |
| Şekil 5.4: Düşme sınıf verisi için aktivasyon haritaları. | 42 |
| Şekil 5.5: İstenmeyen bileşenler barındıran düşme sınıf verisi için aktivasyon haritaları. | 42 |
| Şekil 5.6: 40dB için darboğaz özneliklerinin kıyaslanması. | 43 |
| Şekil 5.7: 7 Sınıf için validasyon performansı. | 44 |
| Şekil 5.8: 12 Sınıf için validasyon performansı. | 44 |

ÇİZELGE LİSTESİ

| | <u>Sayfa</u> |
|--|--------------|
| Çizelge 1.1: Radar sınıflandırma çalışmalarında kullanılan veri seti boyutları (μ D mikro-Doppler'i temsil etmektedir). | 4 |
| Çizelge 5.1: AE için parametre optimizasyon çizelgesi. | 37 |
| Çizelge 5.2: CNN için parametre optimizasyon çizelgesi. | 38 |
| Çizelge 5.3: CAE için parametre optimizasyon çizelgesi. | 38 |
| Çizelge 5.4: SVM için elde edilen karışıklık matrisi (ortalama doğruluk %76.9). | 39 |
| Çizelge 5.5: AE için elde edilen karışıklık matrisi (ortalama doğruluk %84.1). | 39 |
| Çizelge 5.6: CNN için elde edilen karışıklık matrisi (ortalama doğruluk %90.1). | 40 |
| Çizelge 5.7: CAE için elde edilen karışıklık matrisi (ortalama doğruluk %94.2). | 40 |
| Çizelge 5.8: 40 dB verileriyle 7 sınıf probleminde DivNet için elde edilen karışıklık matrisi (ortalama doğruluk %97). | 43 |
| Çizelge 5.9: 40 dB verileriyle 12 sınıf probleminde DivNet için elde edilen karışıklık matrisi (ortalama doğruluk %96). | 43 |
| Çizelge 5.10: Tüm yöntemler için 7 sınıfta test doğruluğu. | 44 |
| Çizelge 5.11: Tüm yöntemler için 12 sınıfta test doğruluğu. | 44 |
| Çizelge 5.12: Tüm yöntemler için hesaplama karmaşıklığı değerleri | 45 |

KISALTMALAR

| | |
|----------------|--|
| DNN | : Derin Yapay Sinir Ağları |
| MOCAP | : Hareket Yakalama |
| DCNN | : Konvolusyonel Derin Yapay Sinir Ağları |
| SAR | : Sentetik Açıklıklı Radar |
| STFT | : Kısa Zamanlı Fourier Dönüşümü |
| FFT | : Hızlı Fourier Dönüşümü |
| CVD | : Cadence Hız Diyagramı |
| DCT | : Ayrık Kösinüs Dönüşümü |
| LPC | : Öngörülü Kodlama Katsayısı |
| SBE | : Sıralı Geri Eleme |
| SVM | : Destek Vektör Makinası |
| RF | : Rastgele Orman Sınıflandırıcısı |
| Xgboost | : Ekstrem Gradyan Yükselticisi |
| RBF | : Radyal Taban Fonksiyonu |
| NN | : Yapay Sinir Ağı |
| RCS | : Radar Kesit Alanı |
| SGD | : Stokastik Gradyan İnişi |
| AE | : Oto-Kodlayıcı |
| CAE | : Konvolüsyonel Oto-Kodlayıcı |
| SNR | : Sinyal gürültü oranı |

SEMBOL LİSTESİ

Bu çalışmada kullanılmış olan simgeler açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler Açıklama

| | |
|------------|--|
| f_D | Doppler kayması |
| f | Gönderilen dalganın merkez frekansı |
| v | Hareketlinin hızı |
| c | Işık hızı |
| f_0 | Yayın frekansı |
| λ | Dalga boyu |
| t | Zaman |
| $R_{t,i}$ | Hedefin zamana bağlı menzili |
| $a_{t,i}$ | Genlik |
| G | Anten kazancı |
| P_i | Verici gücü |
| σ_i | Her bir nokta hedef için RCS |
| L_s | Sistem kayıpları |
| L_a | Atmosferik kayıplar |
| a_0 | Sabit kosinüs terimi |
| w | Sinyalin temel frekansı |
| n | Harmonik terimler |
| φ | Aktivasyon fonksiyonu |
| η | Öğrenme oranı |
| J | Masraf fonksiyonu |
| σ | Aktivasyon fonksiyonu |
| θ | Ağ ağırlıkları ve yanlılıkları |
| β | Seyreklik oranı |
| $1\{.\}$ | İndikatör fonksiyonu |
| M | Aktivasyon haritası |
| f_m | m. konvolüsyonel filtre |
| d | Derinlik |
| s_l | Konvolüsyonel filtrenin uzamsal boyutu |
| * | 2 boyutlu konvolüsyon |
| S_n | Gürültü eklenmiş spektrogram |

1. GİRİŞ

1.1 Problem Tanımı ve Motivasyon

Radar (Radio Detection and Ranging) sistemleri, kızılötesi ve optik gibi diğer algılama sensörlerine göre barındırdığı bariz avantajlar sebebiyle hedef tespiti, hedef takibi ve sınır güvenliği gibi askeri amaçlarla sıkça kullanılmaktadırlar. Aktif bir sensör olan radar sistemi, her türlü hava koşulunda, gece gündüz fark etmeksizin operasyonel kalabilmekte ve uzak mesafelerden hedef tespiti yapabilmektedir.

Radar sistemleri, özellikle insan hareket tanıma problemi için büyük önem arz etmektedirler. Nitekim insan hareket tanıma problemi sınır kontrolü [?] [?], yaya tanıma problemi [?], bakıma muhtaçlar için düşme tespiti [?] gibi konular için önem arz etmektedir. Yakın zamanda gelişen teknoloji sayesinde, mikro-Doppler analizi ile çeşitli insan hareketlerinin tespitinin yapılabildiği gösterilmiştir. Radar sistemlerinin küçülmesi ve ucuzlaması da beraberinde kapalı alan gözetim sistemlerinin radar ile yapılmasının önünü açmıştır. Özellikle yakın zamana kadar bakıma muhtaç insanlar için optik görüntüleme teknolojileri veya giyilebilir sensör tarzı cihazlar kullanılsa da, ışık kaynağına ihtiyaç duymama, mahremiyeti bozmama gibi çeşitli avantajlar radar sensörünü bu problem için uygun kılmaktadır.

Radar, mikro-Doppler etkisi kullanımı ile hareketli nesnelere ait çeşitli karakteristik özelliklerin anlaşılmasını sağlamaktadır. Özellikle insan hareketinin sınıflandırılması için literatürde birçok yöntem önerilmiştir. Ancak literatürde önerilmiş yöntemler birbirine benzer insan hareketlerini sınıflandırmak için yetersiz kalmaktadır. Örneğin evde yalnız yaşayan yaşlı bir insanın düşmesi ile sandalyeye oturması arasındaki farkı, geleneksel yaklaşım (el işçiliği ile öznitelik çıkartıp sınıflandırıcı eğitime) ile çözmenin oldukça zor olduğu görülmektedir [?].

Derin yapay sinir ağları (DNN) yakın zamanda ortaya çıkmış oldukça güçlü bir tekniktir. DNN'ler öznitelikleri veriden otomatik olarak çıkarttıkları için bilgisayarlı görü, doğal dil işleme ve ses işleme konularında devrimsel sonuçlar elde edilmiştir. DNN'lerin bu başarısı, radar komünitesinin de dikkatini çekmiş ve son yıllarda mikro-Doppler tabanlı dron tanıma problemi [?] [?], insan hareketi sınıflandırma [?]-[?]-[?]-[?]-[?] ve işaret tanıma [?]-[?] problemleri için kullanılmaya başlanmıştır.

Ancak DNN'lerin de kendilerine has problemleri vardır. DNN'ler yüksek entropik kapasiteleri sebebiyle verilen girdilerden, girdi sınıflarını genelleyen öznitelikler öğrenebilmek için çok fazla sayıda etiketli veriye ihtiyaç duymaktadırlar. Çok yüksek miktarda radar verisi toplamak masraflı olduğundan standart DNN modelleri ile radar mikro-Doppler hareket sınıflandırma problemine çözüm getirmek zordur. Bu tezin ana katkısı, mikro-Doppler sınıflandırma problemi için az sayıda veri olduğu durumda DNN modellerinin nasıl eğitileceği problemine önerilen çözümdür.

1.2 Mikro-Doppler Etkisi

Doppler etkisi, bir dalga kaynağının gözlemciye olan görelî hareketine bağılı olarak, dalganın frekansında gözlemlenen deęişimi açıklayan fenomendir. 1842 yılında isim babası olan Christian Doppler tarafından tanımlanmıştır [?]. Doppler etkisinin en bilindik örneęi olarak 1929 yılında Edwin Hubble'ın gözlemi verilebilir. Hubble, çeşitli galaksilerin tayfinin kırmızı renkte olduğunu gözlemlemiştir: Kırmızı ışık, gözlemlenebilir spektrumda dalga boyu en büyük dolayısıyla frekansı en düşük ışık olduğundan bunun tek bir açıklaması olabilirdi: Bu galaksiler dünyadan uzaklaşmaktadır.

Radar sistemleri de gözlemlenen hedefe dair çeşitli ölçümler elde edebilmek için gönderdikleri elektromanyetik dalganın maruz kaldığı doppler etkisinden faydalanmaktadır. Radardan gönderilmiş, c hızı ile ilerleyen bir elektromanyetik dalganın, c 'den çok daha düşük v hızı ile ilerleyen bir hedeften yansımısıyla uğradığı toplam doppler kayması

$$f_D = -f \frac{2v}{c} \quad (1.1)$$

olarak tanımlanmaktadır. Bu ifadede f_D doppler kaymasını, f gönderilen dalganın merkez frekansını, v hareketlinin hızını ve c ışık hızını belirtmektedir. Hedefin radara yaklaşması durumunda doppler kayması pozitif olmakta, aksi durumda ise negatif olmaktadır.

Hedef eđer hareketli parçalar barındırıyor ise, bu hareketli parçalar ana frekans kayması üzerine eklenmiş mikro frekans kiplemelerine sebebiyet vermektedirler [?]. Örneğin bir insan hareketi ele alındığında, insan yürüyüşünün doğası gereęi kol ve bacak salınımları olacak ve gövdenin yaratacağı ana doppler kaymasına ek olarak, ana doppler kaymasının yan bantlarında frekans kaymalarına sebebiyet verecektir. Bu mikro kiplemelerin yarattığı etkiye mikro-Doppler etkisi denilmektedir. Mikro kiplemeler hareketten harekete farklılık göstereceğinden, farklı hareketlerin sınıflandırılması amacıyla kullanılabilir.

1.3 Geleneksel Yöntem

Geleneksel yöntem ile mikro-Doppler verilerinin sınıflandırılması için ilk olarak öntanımlı öznitelikler çıkartılır. Bu öznitelikler yansıma sinyalinin zaman serisi formu üzerinden veya zaman-frekans analizi yapılarak elde edilmiş spektrogram üzerinden çıkartılabilir. Özniteliklere örnekler olarak, spektrogram üzerinden çıkartılan fiziksel öznitelikler (Doppler bant genişlięi, alt ve üst zarflar vb.) [?], dönüşüm tabanlı öznitelikler (ayrık kosinüs dönüşümü) [?], ses işleme tabanlı öznitelikler (mel-frekans katsayıları [?], doğrusal öngörölü kodlama [?]) ve boyut azaltma (temel bileşen analizi [?] ve tekil deęer ayrışımı [?]) verilebilir. Daha sonra çıkartılmış olan bu özniteliklerin bir kısmı öznitelik seçim algoritmaları kullanılarak seçilir (opsiyonel olarak) ve sonrasında el ile çıkartılmış bu öznitelikler gözetimli bir sınıflandırıcının eğitilmesi için kullanılarak veriler sınıflandırılır.

1.4 Güncel Literatür ve Önerilen Yöntem

Son zamanlarda DNN'ler, milyonlarca verinin kolayca elde edilebildiği bilgisayarlı görü, doğal dil işleme ve ses işleme alanlarında çığır açmıştır. Özellikle bilgisayarlı görü alanında yıllık düzenlenen ImageNet [?] yarışması sayesinde obje tanıma üzerine birçok farklı Derin Konvolüsyonel Yapay Sinir Ağı (DCNN) yapısı geliştirilmiştir. ImageNet veri seti toplamda 1000 farklı sınıf içeren 1.5 milyon Kırmızı Yeşil Mavi (RGB) görüntüden oluşmaktadır. Derin öğrenmenin bu kadar popüler olmasının başlangıcı sayılabilecek bir DCNN modeli olan Alexnet [?] 8 katmanlı bir DCNN önererek yayınlandığı yıl olan 2012'ye kadar görülmemiş bir performans elde etmiştir. Bunun ardından, 2014 yılında 22 katmanlı bir DCNN modeli olan GoogleNet [?] ve 16 katmanlı bir DCNN modeli olan VGGNet [?] modelleri sırasıyla 1. ve 2. liği paylaşmışlardır. Sonrasında 2015 yılında, Microsoft tarafından o zamana kadar görülmuş en derin DCNN modeli olan ResNet modeli ImageNet'i kazanmıştır. Bu model 152 katmanlı bir DCNN önermektedir [?]. Yıllar geçtikçe önerilen mimariler derinleşmiş ve ImageNet üzerine alınan başarımlar artmıştır.

Ancak DNN'ler derinleştikçe, iyi bir genelleme öğrenmeleri için gereken veri sayısı da artmaktadır. Radar ile veri toplamak zaman ve iş gücü bakımından çok fazla kaynak gerektirmektedir ve ImageNet skalasında bir veri seti oluşturmak oldukça masraflıdır. Örnek verilecek olursa, literatürde önerilmiş mevcut veri kümeleri, genellikle mikro-Doppler için yüzlerce örnek, SAR görüntüleri için binlerce örnek seviyesindedir (Çizelge 1.1). Bu tezin amacı olan insan hareket sınıflandırma problemi oldukça zor bir problem olarak ortaya çıkmaktadır. Nitekim insanların boyları, vücut yapıları, hareket etme stilleri farklılık göstermekte ve hareketlerin genellenebilmesini zorlaştırmaktadır. İnsan hareketlerinin genellenebilmesi için eğitim verilerinin mümkün olduğunca çok hareket varyasyonu içermesi gerekmektedir, ancak bunu gerçekleştirmek oldukça masraflıdır.

Bu durum açıkça bir araştırma problemini doğurmaktadır: Az sayıda veri ile DNN eğitmek için nasıl bir yol izlenmelidir? Tipik olarak, DNN'lerin ağırlıkları rastgele sayılarla başlatılmaktadır ve öğrenme işlemi sonrası belli değerlere yakınsamaktadırlar. Fakat, DNN'lerin objektif fonksiyonu karmaşık problemler için çoğunlukla konveks değildir, yani bir sürü lokal minimum içermektedir. Buna bağlı olarak gradyan tabanlı öğrenme algoritmalarının bu lokal minimumlara takılma olasılığı yüksektir. Bu durumda az sayıda veri ile eğitildiklerinde DNN'ler genel olarak evrensel yakınsayıcılar olduklarından, verilen az sayıda veriyi hemen ezberleyecek, dolayısıyla görmedikleri bir test verisi için problemi genellememiş olacaklar ve düşük performans göstereceklerdir.

Bunun yanında DNN ağırlıklarını rastgele başlatmaktansa çeşitli yöntemler önerilmiştir. İlk yaklaşım, 2006 yılında Hinton tarafından önerilen gözetimsiz ön-eğitim yaklaşımı bir ağırlık başlatımı sağlamaktadır [?]. Gözetimsiz ön-eğitim yöntemi ile bir kodlayıcı ve çözümleyiciden oluşan bir özkodlayıcı yapısı oluşturularak, ağgözlü bir şekilde katmandan katmana eğitim sağlanabilir. Sonrasında kod çözücü networkten çıkartılır ve ön-eğitim ile ağırlıkları başlatılmış kodlayıcı kısım etiketli veri ile ince ayar yapılarak eğitilir. Bu yaklaşım DCNN yapısı için de bir Konvolüsyonel özkodlayıcı (CAE) kurularak uygulanabilir [?] [?].

Çizelge 1.1: Radar sınıflandırma çalışmalarında kullanılan veri seti boyutları (μ D mikro-Doppler'i temsil etmektedir).

| Uygulama | Referans | Sınıf Sayısı | Veri Sayısı |
|----------------|-------------------------|--------------|-------------|
| insan μ D | Jokanovic, et al. [?]] | 4 | 120 |
| insan μ D | Kim ve Moon [?]] | 4 | 160 |
| dron μ D | Mendis, et al. [?]] | 3 | 210 |
| işaret μ D | Kim ve Toomajian [?]] | 10 | 500 |
| insan μ D | Seyfioglu, et al. [?]] | 12 | 864 |
| araç radarı | Lombacher, et al. [?]] | 10 | 3,397 |
| dron μ D | Kim, et al. [?]] | 5 | 60,000 |
| SAR | Li ve Wang [?]] | 4 | 66,120 |

DNN ağırlıklarının başlatmanın bir diğer yolu ise aktarımlı öğrenim [?]] yöntemini kullanılmaktadır. Aktarımlı öğrenim, ağırlıklarının başlatılması için, farklı ancak ilintili bir problemten gelen verilerle eğitilmiş bir modelin, ilgili problem için ince ayar ile eğitilmesi durumudur. Çok basit bir örnek olarak aktarımlı öğrenim köpekleri kedilerden ayırt etmeyi öğrenmiş bir modelin, farklı köpek türlerinin sınıflandırılması problemine uyarlanması olarak özetlenebilir.

Aktarımlı öğrenim sentetik açıklıklı radar (SAR) görüntü sınıflandırma [?]] ve hareketli hedef tanıma [?]] uygulamalarında önerilmiştir. Yakın zamanda, Park [?]], 5 değişik yüzme hareketini az sayıda veri ile aktarımlı öğrenim kullanarak sınıflandırmıştır ve sonuçların rastgele başlatılan ağa göre daha iyi olduğu vurgulanmıştır. Şüphesiz ki ImageNet gibi devasa bir veri seti üzerinde eğitilmiş ve iyi performans sağlayan modeller, kenar, hat, köşe gibi temel öznitelikleri çıkartmayı öğrenmiş ve mikro-Doppler sınıflarını ayırt etmek için bunları kullanabilmektedirler. Ancak RF ve optik veriler arasındaki bariz fark bu yaklaşımın performansını kısıtlamaktadır. Nitekim ImageNet veri seti doğal objeler içerdiğinden mikro-Doppler imzasına benzer bir veri içermektedir.

Peki veri sayısı az olduğu durumda DNN eğitebilmek için gözetimsiz ön-eğitim veya aktarımlı öğrenim uygulamak dışında ne yapılabilir? Hem eğitim verisi sayısını hem de sınıf içi varyasyonları arttırmanın bir yolu simüle edilmiş verileri kullanmaktır. Gerçek mikro-Doppler ölçümlerini sınıflandırmak için simüle imza kullanımı ilk olarak 2015 yılında Karabacak tarafından önerilmiştir [?]]. Bu çalışmada gerçek radar ölçümü ile elde edilmiş mikro-Doppler verilerini sınıflandırmak için Carnegie Mellon Üniversitesi (CMU) hareket yakalama (MOCAP) verileri ile eğitilmiş bir sınıflandırıcının kullanılmasını önerilmiştir. Simülasyonlar ayrıca SAR [?]] ve yüksek çözünürlüklü menzil profili [?]] verilerinin sınıflandırılması için de kullanılmıştır. Daha yakın bir tarihte, Kinect gibi düşük maliyetli cihazlar insan yürüyüşünü simüle eden veriler üretmek için kullanılmaya başlanmıştır. Literatürde simülasyon verileriyle eğitilmiş modeller üzerinden hareket tanıma problemi üzerine birçok çalışma mevcuttur: [?]] - [?]] [?]] [?]] [?]] [?]] [?]] ayrıca düşüş tespiti için de simülasyon verilerinin kullanıldığı çalışmalar vardır [?]] - [?]] [?]].

Bununla birlikte, Kinect simülasyonları gerçek radar verilerinde olduğu gibi insanlar üzerinden veri toplanmasını gerektirdiği için veri sayısı yine sınırlı olmaktadır. An-

cak, Kinect ile elde edilmiş simülasyon verileri üzerinden, sınıf içi varyansı maksimize edecek bir yöntem ile (insan hareketleriyle tutarlı olacak şekilde) büyük bir veri seti oluşturulabilir. Bu veri çeşitlendirme işlemi Kinect tabanlı imzaların çeşitli parametrelerinin değiştirilmesi ile elde edileceği için, insan hareketlerini tam doğruluk ile modelleyemeyecektir. Ancak DNN'ler veri üzerinden zaten bir genelleme öğrendikleri için, yeterince büyük bir veri seti ile eğitilirse simülasyon verilerindeki ufak hataları ezberlemeyecekler ve sınıfları ayırt etmek için gerekli öznelikleri öğrenmeye başlayacaklardır.

1.5 Tez Kapsamı

Tez kapsamında, birbirine oldukça benzeyen 12 farklı insan hareketinin mikro-Doppler imzaları kullanılarak sınıflandırılması için DNN ağırlıklarının başlatılması üzerine çeşitli teknikler araştırılmıştır. Bu neticede öncelikle bir sürekli dalga radarı ile 11 farklı kişiden toplamda 1007 veri toplanarak bir mikro-Doppler veri seti oluşturulmuştur.

Oluşturulan veri seti üzerinde ölçüt noktası (baseline) sonuçlar oluşturması açısından geleneksel yöntem kullanımı ile derinlemesine bir analiz yapılmış, bu veri seti için el işçiliği ile çıkartılmış öznelikler ile alınabilecek en yüksek başarımları gösterilmiştir.

Yapay sinir ağları ve DNN'ler üzerine derinlemesine bir analiz yapılmış, DNN'lerin az sayıda veri ile eğitilmesinin getirdiği limitasyonlar incelenmiş ve az sayıda veri ile DNN eğitimi konusu üzerine literatürde önerilmiş yöntemler üzerine çalışılmıştır.

DNN'lerin genelleme yapabilmek için gereksinimi olan çok fazla veri ile eğitim probleminin çözümü için yenilikçi bir yöntem önerilmiştir. Bu kapsamda 7 sınıf içeren simülasyon verisi ile eğitilmiş bir derin yapay sinir ağı, 12 sınıf içeren gerçek veriler üzerine 'ince ayar' ile eğitilmiştir.

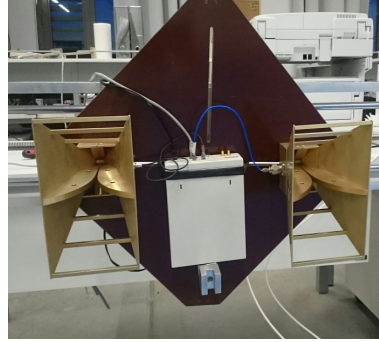


2. TEZ KAPSAMINDA OLUŞTURULMUŞ VERİ SETLERİ

Bu çalışmada 2 ayrı veri seti oluşturulmuştur: Deneysel veri seti ve Kinect tabanlı simülasyon veri seti. Deneysel veri seti, laboratuvar ortamında toplanmış gerçek radar ölçümlerinden oluşmaktadır ve toplamda 12 hareket için 1007 veri içermektedir. Kinect tabanlı simülasyon veri seti ise 7 insan hareketi için 32000 veriden oluşmuştur ve insan hareketlerindeki hızdan, boydan ve hareket stilinden kaynaklı değişiklikleri modeller.

2.1 Deneysel Veri Seti

Bu tez kapsamında oluşturulmuş veri seti NI-USRP 2922 marka bir yazılım-tabanlı radyo platformunun sürekli dalga radarı olarak programlanması ile elde edilmiştir. Ölçümler 1-5 metre uzunluğunda bir düzlemde kapalı laboratuvar ortamında alınmıştır. Radar sistemi 4 GHz merkez frekansına sahiptir ve alıcı verici yapısı olarak iki adet SAS-571 anten kullanılmıştır. Antenler 48°'lik azimutsal genişliğe sahiptir. Radar sistemi yerden 1 metre yukarıda olacak şekilde monte edilmiştir. Ölçümler, hedeflerin radara 0° açı ile (tam karşısından) yaklaşması ile alınmıştır. Radar sistemi genel olarak Şekil 2.1 üzerinden görülebilir.



Şekil 2.1: Radar donanımının konfigürasyonu.

Veri seti toplamda 11 ayrı insandan toplanmış 1007 veriden oluşmakta ve 12 farklı sınıf içermektedir. Her bir sınıf için toplanmış veri sayıları şu şekildedir: Yürüme 71, koşma 72, emekleme 74, topallama 104, baston 123, düşme 53, tekerlekli sandalye 149, koltuk değneği 74, oturma 50, yürüteç 121, sandalyeden düşme 60, sürünme 56. Hareketlere ait örnek spektrogram görüntüleri Şekil 2.2 üzerinde gösterilmiştir. Hareketler aşağıdaki tanımlar üzerinden gerçekleştirilmiştir:

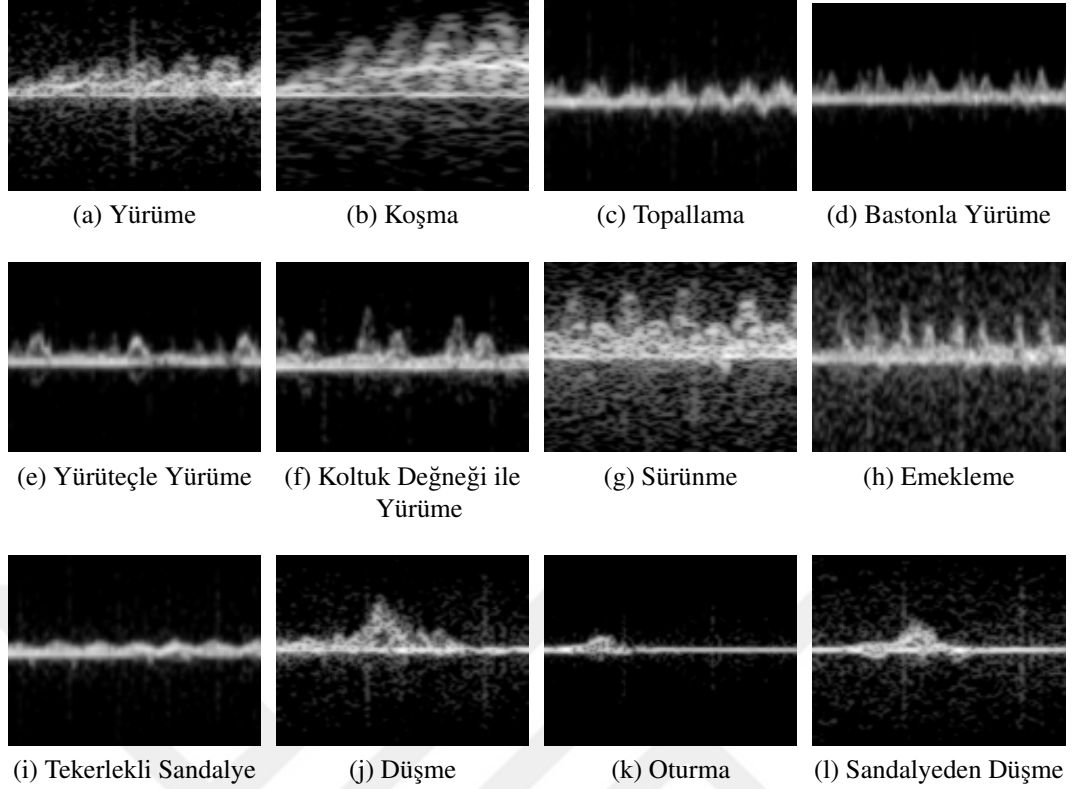
1. Yürüme – orta hız, 2 kol yürürken tamamen sallanmakta.
2. Koşma – yüksek hız, kollar dirsekten kırılmış, sallanma kısıtlı.
3. Topallama – sol ayak sağ ayağın gerisinden sürüklenmekte.
4. Baston ile yürüme – Tek kol ile kontrol edilen metal bir baston yardımı ile yavaşça yürüme.
5. Yürüteç ile yürüme – iki tekerlekli metal bir yürüteç yardımı ile yürüme, iki kolun da yürüteçi tutmakta dolayısıyla kol salınımı yok.
6. Koltuk değneği ile yürüme – iki metal değnek kullanılmakta ve hareket esnasında tek bacak dizden bükülmekte.
7. Emekleme – Eller ve dizler yere değerken yavaşça ilerleme.
8. Sürünme – Karın yere değecek şekilde askeri stilde sürünme hareketi.
9. Tekerlekli sandalye kullanımı – İki kol ile tekerlekler ilerletilerek kullanılmakta. Ayaklar tamamen sabit.
10. Düşme – Bir objeye takılıp hızla yere düşme.
11. Oturma – Sandalyeye kendini bırakarak hızla oturma.
12. Sandalyeden düşme – Sandalyeden yanlamasına yere düşüş.

2.2 Aktarımlı Öğrenme İçin Simülasyon Veri Setinin Oluşturması

ImageNet gibi mikro-Doppler'den tamamen alakasız bir veri seti üzerine eğitilmiş bir ağın, az sayıda Mikro-Doppler verisi kullanılarak ince ayar ile eğitilmesi sonucunda bile oldukça yüksek sınıflandırma başarımlarına ulaşabildiği görülmüştür. Bu durum şu iki soruyu beraberinde getirmektedir:

1. Aktarımlı öğrenme, çok sayıda simülasyon Mikro-Doppler verisi ile eğitilmiş bir model üzerinden uygulanırsa, aktarım alanının problem ile aynı olması bir performans artışı sağlar mı?
2. Yalnızca belli hareketlere ait mikro-Doppler imzalarını görmüş bir modelin görmediği sınıfları ayırt edebilecek şekilde ince ayar ile eğitilebilir mi?

Bu iki soruya yanıt aramak için, Villanova Üniversitesi Radar Görüntüleme Laboratuvarı ile ortak bir çalışmamızda oluşturulmuş olan, büyük miktarda mikro-Doppler verisi içeren bir simülasyon veri tabanı kullanılmıştır. Bir sonraki Bölümde bu veri setinin nasıl oluşturulduğu detaylı bir şekilde anlatılmıştır.



Şekil 2.2: İncelenen insan hareketleri için spektrogram örnekleri.

2.2.1 Kinect tabanlı simülasyon veri seti

İnsan mikro-Doppler imzalarını simüle etmek için literatürde önerilmiş iki yaklaşım vardır [?]: Kinematik modelleme ve hareket yakalama (MOCAP) tabanlı animasyon. Her iki yöntem de ana fikir olarak insan vücudunu, sonlu sayıda noktasal hedefle modellemekte ve bu noktasal hedeflerden ayrı ayrı gelen radar dönüşlerini, radar menzil denklemine göre hesaplayıp toplamak üzerine kurulmuştur [?]. Radar literatüründe en yaygın kullanılan kinematik model Boulic-Thalman modelidir [?]. Bu model, insan hareketini, 17 farklı eklemin zamana bağlı değişimiyle açıklayan, deneysel bir tabanı olan bir dizi denklemden oluşmaktadır. Boulic modelin en büyük dezavantajı ise yalnızca yürüyüş hareketini modelleyebiliyor olmasıdır. Buna bağlı olarak diğer ritmik olmayan ve yardıma dayalı (bastonla yürüme vb.) yürüme hareketlerinin Boulic model kullanılarak ile simüle edilmesi oldukça zordur. Bu nedenle, yakın zamanda, MOCAP tabanlı animasyonlar, mikro Doppler simülasyonları için yaygınlık kazanmıştır.

Tez kapsamında, insan mikro-Doppler imzalarının simülasyonu için gerekli olan insan eklemlerinin zamanla değişen koordinat bilgilerini yakalamak için Kinect sensörü kullanılmıştır [?]. İlk olarak, insan vücudundan gelen radar geri dönüşü, çeşitli vücut parçalarını temsil eden sonlu sayıda (K) noktalı hedeften yansıyan sinyallerin toplamı olarak gösterilir ve bu tez kapsamında K 20 olarak belirlenmiştir. Kinect ölçümleri, genellikle radardan elde edilen zamanla değişen menzil ölçümleri doğrultusunda alınmıştır. Matematiksel olarak, bir sürekli dalga radarı için K noktalı hedeflerden gelen dönüş sinyal şu şekilde ifade edilir:

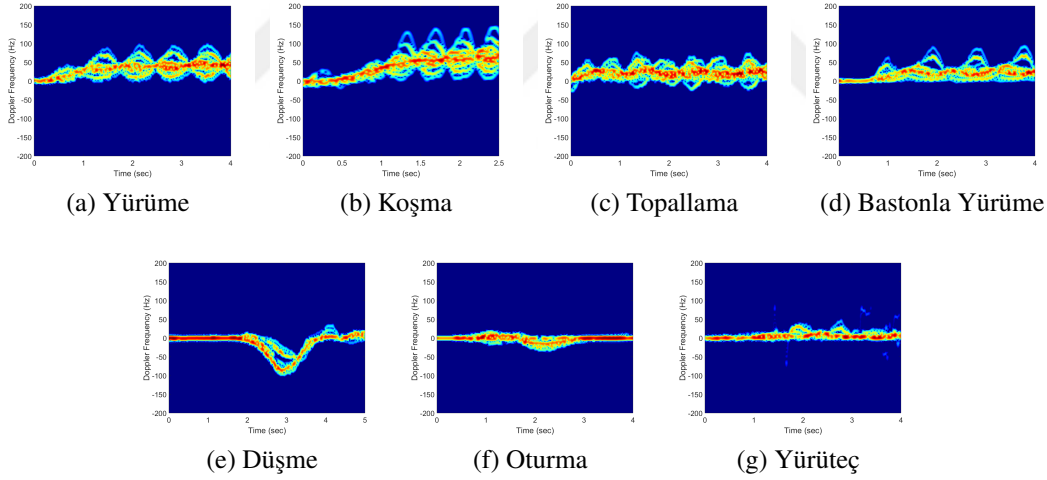
$$s_h(t) = \sum_{i=1}^K a_{t,i} e^{-j[(2\pi f_0)t + \frac{4\pi}{\lambda} R_{t,i}]} \quad (2.1)$$

burada f_0 yayın frekansını, λ dalga boyunu, t zamanı, $R_{t,i}$ her hedefin zamana bağlı menziline ve son olarak $a_{t,i}$ radar menzil denkleminde hesaplanan genlik değerini belirtmektedir ve aşağıdaki gibi hesaplanmaktadır:

$$a_{t,i} = \frac{G\lambda\sqrt{P_i\sigma_i}}{(4\pi)^{1.5}R_{t,i}^2\sqrt{L_s}\sqrt{L_a}}. \quad (2.2)$$

Burada, G anten kazancını, P_i verici gücünü, σ_i , her bir nokta hedef için radar kesit alanını (RCS) ve L_s ve L_a sırasıyla sistem ve atmosferik kayıpları temsil etmektedir. Kafa bölgesinin RCS'i bir küreyle, gövde ve uzuvlar ise bir elipsoit ile temsil edilmektedir.

Kinect verisi üzerinden menzil kestirimi yapıldıktan sonra, Denklem (2.1) merkez frekansı, bant genişliği, örnekleme frekansı vb. gibi herhangi bir parametre için hesaplanabilir. Son olarak, hesaplanan radar verileri için zaman frekans analizi Denklem (2.5) üzerinden yapılabilir. Bu işlem toplam 7 farklı insan hareketini simüle etmek için tekrarlanmıştır ve insan hareketlerine ait görseller Şekil 2.3 üzerinden görülebilir.



Şekil 2.3: 15 GHz CW radar için üretilmiş Kinect Tabanlı Mikro-Doppler imzaları.

2.2.2 Simüle Mikro-Doppler imzaları için farklılaştırma metodolojisi

Kinect tabanlı radar mikro-Doppler simülöründe, Kinect sensöründen elde edilen 17 eklemin 3 boyutlu koordinat ölçümleri kullanılmıştır. Bu koordinat bilgilerini değiştirerek, insandan insana değişen boy, hız ve bireysel yürüyüş farklılıklarını simüle etmek mümkündür.

2.2.2.1 Boy ve hız modifikasyonları

Kinect tabanlı radar simülatörü ile, zamanla değişen eklem konumu verilerini farklı eksenler boyunca ölçeklendirerek simüle edilen hareketlinin boyu ve hızı değiştirilebilir. Örneğin, 3 boyutlu koordinat uzayında x ve y eksenleri değiştirilmeksizin z eksenini boyunca yapılan ölçeklendirme, simüle edilen hareketlinin boyunu değiştirmektedir. Bu değişim hareketlinin diğer vücut parçalarına da etki etmekte, dolayısıyla tekil nokta yansıtıcılarının RCS'ini değiştirmektedir. Buna bağlı olarak değişen RCS, alıcıya gelen güç miktarına etki edecektir. Hareket farklılaştırma metodolojisinin ilk etabında ölçeklendirme yalnızca z eksenini boyunca yapıldığından, adım hızı, genel hız ve yürüyüş stili etkilenmez. Bununla birlikte, gerçek dünya incelendiğinde, vücut kütlesi, esneklik ve orantılılık gibi vücut yapısı faktörleri eşit kabul edildiği zaman, uzun boylu bir kişinin genellikle kısa bir kişiden daha hızlı yürüyeceği ya da koşacağı aşırıdır. Bu durumu simülasyona yansıtma adına, hareketlinin z eksenindeki değerine (boyuna) bağlı olarak y eksenini de ölçeklendirilmiştir.

Aynı hareketlinin, farklı boylardaki animasyonundan alınan anlık ekran görüntüleri Şekil 2.5(a) ve Şekil 2.5(b) üzerinden görülebilir. Şekil 2.5(a)'da hareketlinin boyu 155 cm olarak belirlenmiştir. Bu değer Şekil 2.5(b)'de 190 cmdir. Bu hareketlilerin mikro-Doppler imzaları sırasıyla Şekil 2.5 (e) ve (f)'de gösterilmiştir. Mikro-Doppler imzalarına bakıldığında, uzun boylu hareketli için sağ ve sol ayaktan gelen yansımaların daha belirgin olduğu görülmektedir. Bunun sebebi olarak bacak uzunluğunun fazla olması gösterilebilir. Ayrıca, hareketlinin boyundan kaynaklı hız değişimi de spektrogram üzerinden belirgin bir şekilde görülmektedir.

Mikro-Doppler imzasını büyük ölçüde etkileyen ikinci bir parametre, hareketlinin hızıdır. Boulic-Thalmann modelinde hareket hızı, modelin bir parametresini (döngü süresi) basitçe değiştirerek ayarlanabilir. Parametrelerin geri kalanı buna göre farklı vücut parçaları arasındaki kinematik ilişkiler yoluyla değişir. Fakat, bu işlemin Kinect tabanlı MOCAP verisiyle yapılması, hareketlinin ölçülen pozisyonunun halihazırda bir hız bilgisi içeriyor olması sebebiyle zordur. Buna bağlı olarak y ekseninde bir işleme daha ihtiyaç vardır: Eğer basitçe ham Kinect verisinin örnekleme frekansı değiştirilirse, bu değişim adım hızını ve hareketlinin hızını etkileyecektir. Şekil 2.5(c) ve (d)'de iki ayrı hız değeri için animasyonların ekran görüntüleri gösterilmiştir. Aynı zaman aralığı içerisinde, hızlı bir hareketlinin yavaş bir hareketliye göre daha uzun mesafe katettiği açıktır. Mikro-Doppler imzaları da Şekil 2.5(g) ve (h)'de aynı değişkenler için gösterilmiştir. Burada aynı zaman aralığında hızlı hareketlinin uzuvlarının, yavaş hareketlinin uzuvlarına göre daha kısa döngüler içerdiği gözlemlenmektedir.

Simülasyon metodolojisindeki bir diğer önemli husus, farklı hareketlere ait imzaların örtüşmesine neden olan uç durumların nasıl engelleneceğidir. Örneğin hızlı yürüme ile yavaş tempoda koşma birbirine oldukça benzer iki imzadır ve sınıflandırıcıya karışıklık oluşturacaktır. Bu iki hareketin varyasyonları muhtemelen hareket hızının değişimi sonucunda aynı frekans bantlarında bulunabilecektir. Bu nedenle, bu gibi istenmeyen çakışmaları önlemek için, 7 farklı insan hareketi için en yüksek ve en yavaş hızları ve ilgili Doppler frekansları (15 GHz merkez frekansı) belirlenmiş ve çeşitlendirilmiş imzalar buna göre sınırlandırılmıştır.

2.2.2.2 Eklem parametrelerinin belirlenmesi

Çeşitlendirme metodolojisinin son adımında sol ve sağ bacak, sağ ve sol kol ve baş gibi bireysel eklem verilerine odaklanılır. Buradaki temel fikir, farklı eklemlerin Kinect ham z eksen verilerini ayrı ayrı parametrelemektir. Daha sonra, oluşturulan modellerin katsayılarını karıştırarak, sınıf varyasyonları yaratmak mümkündür. Model parametrelerinin sınırlı değişiklikleri, hareketin stilini etkilemektedir.

Eklem parametreleştirilmesi işlemi eğri uydurma modelleri kullanılarak yapılabilir. Bu kapsamda sinüzoidal, Fourier serileri (harmonik), polinom, doğrusal interpolasyon gibi birkaç eğri uydurma modeli denenmiştir. Eklem verilerinin periyodik doğası göz önüne alındığında ve oluşturulan modelin iyiliği incelendiğinde, Fourier serisi bu problem için en uygun parametreleme modeli olarak belirlenmiştir. Bu modelin ayrıca düşme ve oturma gibi periyodik olmayan hareketlerin parametrelerinin belirlenmesi hususunda da etkili olduğu görülmüştür. Fourier serisi modeli verilen Kinect verilerini sinüs ve kosinüs fonksiyonlarının bir toplamı olarak tanımlar. Ortaya çıkan model trigonometrik formda aşağıdaki şekilde temsil edilebilir:

$$y = a_0 \sum_{i=1}^n a_i \cos(iwx) + b_i \sin(iwx) \quad (2.3)$$

Burada a_0 verinin içerisindeki $i = 0$ durumunda sabit kosinüs terimini modeller, w sinyalin temel frekansdır ve n ($0 < n < 9$) harmonik terimlerini tanımlar. Bu model $2n$ katsayı içermektedir ve bu da kullanılan harmonik sayısı ile doğrudan ilintilidir. Eklem verilerinin altında yatan bilgileri korumak için, sadece n -çift harmonik katsayıları $[(a_1, b_1), \dots, (a_n, b_n)]$ değiştirilmiştir. Değişim işlemi, aynı anda %10'luk menzil değeri içinde olan bir çift değiştirilerek tamamlanmıştır. Metodolojiyi farklı hareketlerin kinematiki ile tutarlı hale getirmek için bazı kısıtlar da eklenmiştir. Örneğin, bir harmoniğin çift katsayı ikilisi farklı eklemler için değiştiğinde, algoritma otomatik olarak değişmiş kol ve bacak uzunluklarını orjinal uzunluklarıyla kıyaslayarak, yapılan bu alterasyonun kinematik olarak mümkün olup olmadığına bakar. Bu simülasyon metodolojisi Şekil 2.4'te özetlenmiştir.

2.2.3 Önerilen metodolojinin validasyonu

Veri çeşitlendirme algoritmasının doğruluğu hem göz ile ekstrem örneklerin değerlendirilmesi hem de görsel kıyaslama metriği olan yapısal benzerlik indeksi (SSI) hesaplanarak incelenmiştir. SSI sınıf içi ve sınıflar arası olarak hesaplanmıştır. Gözle yapılan gözlemler, maksimum ve minimum hız/boy değerleri göz önüne alınarak yapılmış, ekstrem örnekler çeşitlendirilmiş veri setinden çıkartılmıştır. Tüm veriler gözle incelenemeyeceğinden, SSI metriği yardımıyla genel olarak veri setinin sınıf içi ve sınıflar arası benzerlik ve farklılığı incelenmiştir.

SSI bir görsel kıyaslama metriğidir. Bir görselin maksimum kalitede olduğu durumda diğerinin ona ne kadar yakın olduğunu hesaplar. SSI üç ana terimi hesaba katar: Parlaklık terimi, kontrast terimi ve yapısal terim. Genel endeks, üç terimin çarpımsal bir birleşimidir ve aşağıdaki gibi verilir:

$$SSI(x, y) = \frac{2(\mu_x \mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.4)$$

burada x ve y kıyaslanan görselleri, μ_x , μ_y , lokal ortalamaları, σ_x , σ_y , standart sapmaları ve σ_{xy} ise çapraz kovaryans değerini vermektedir. Ayrıca C_1 ve C_2 sırasıyla parlaklık ve kontrast değerleri için regularizasyon sabitlerine karşılık gelir. İlk olarak, SSI, Şekil 2.6(a)'da tasvir edildiği gibi sınıflar arası benzerlik haritasını elde etmek için hesaplanır. Şekil 2.6(a) üzerinden düşme ve oturma sınıfına ait örneklerin SSI değerlerinin oldukça benzer olduğu görülmektedir. Bu durum bu iki hareketin benzer kinematik yapısından kaynaklanmaktadır. Düşme ve oturma hareketleri dışında, yalnızca birkaç örneğin benzerlik taşıdığı görülmektedir.

İkincil olarak, orjinal mikro-Doppler imzaları ile çeşitlendirme algoritmasının ürettiği imzalar arasındaki benzerlikler incelenmiştir. Eldeki kısıtlı sayıda orjinal veriden varyans üretmek için, orjinal ve üretilmiş görseller arasındaki benzerlik 1'den küçük bir değere sahip olmalıdır. Bu değer çok düşük tutulursa, yüksek benzerlik oluşacak ve sınıflandırma başarımı olumsuz yönde etkilenecektir. SSI orjinal görseller ve üretilmiş varyantlar için hesaplanmış ve Şekil 2.6(b) üzerinden yürüme, koşma ve topallama hareketleri için gösterilmiştir. Boy değerinin her iterasyonunda, SSI sınıflara özgü örüntüler üretmekte, bu da oluşturulan varyantların kendi içinde tutarlı olduğunu göstermektedir.

2.2.4 Oluşturulan simülasyon Mikro-Doppler veri seti

Önerilen metodoloji kapsamında toplam 5 farklı kişiden toplanmış 55 Kinect tabanlı MOCAP verisi üzerinden 7 farklı hareket için 32000 adet imza oluşturulmuştur. Hareket sınıfları yürüme, koşma, topallama, düşme, oturma, bastonla yürüme ve yürüteçle yürüme olarak belirlenmiştir.

Algorithm 1 Kinect Data Perturbation Scheme

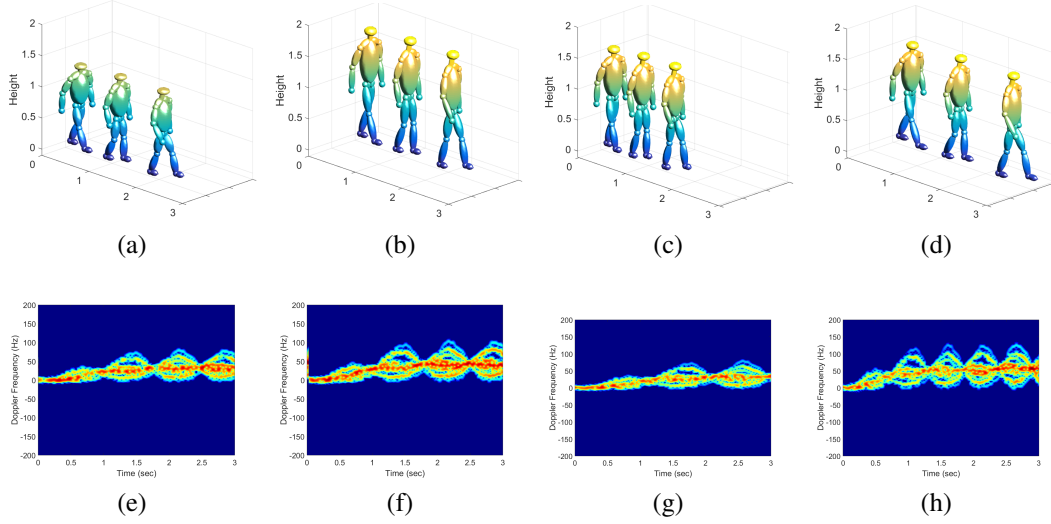
Input: Kinect 3-D Joint Data, height (h_{val}) and speed (s_{val}) coefficients, signal-to-noise ratio (SNR), confidence interval (c), number of harmonics in the parameterization (n), original arm length l_a , original leg length l_l

Output: I , micro-Doppler image

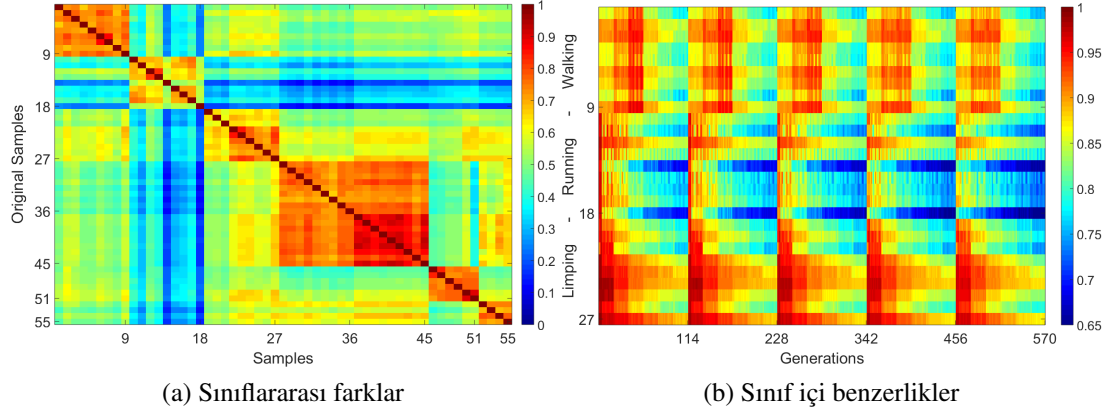
Initialization:
Adjust the Kinect 3-D raw Data

- 1: **for** 1 to $size(h_{val})$ **do**
- 2: Scale all joints positions in y-axes with h_{val}
- 3: **for** 1 to $size(s_{val})$ **do**
- 4: Scale all joints positions in z-axes with s_{val}
- 5: Fourier parameterization for right-left arm, leg and head joint data
- 6: **while** $RMSE < 1$ and $model_P < 1$ **do**
- 7: **for** 1 to n **do**
- 8: Alteration of the n th harmonic's parameters for right and left arm, leg, and head joints within c
- 9: **if** ($ArmLength > l_a$ or $ArmLength < l_a$) **then**
- 10: Use original leg joint data
- 11: **end if**
- 12: **if** ($LegLength > l_l$ or $LegLength < l_l$) **then**
- 13: Use original leg joint data
- 14: **end if**
- 15: Compute micro-Doppler signature, I , using altered data
- 16: **end for**
- 17: **end while**
- 18: **end for**
- 19: **end for**
- 20: **return** I

Şekil 2.4: Kinect verisi için pertürbasyon algoritması.



Şekil 2.5: Tek bir veriden çıkarılmış Kinect tabanlı simülasyon sonuçları: (a) Kısa bir kişi (b) Uzun bir kişi (c) adım hızı yavaş olan bir kişi, (d) adım hızı hızlı olan bir kişi ve bunlara sırasıyla karşılık gelen mikro-Doppler imzaları (e) Kısa (f) Uzun (g) Yavaş, ve (h) Hızlı.



Şekil 2.6: Önerilen imza çeşitlendirme tekniğinin doğrulanması: (a) Sınıflararası farkların kıyaslanması (1-9 Yürüme, 9-18 Koşma, 18-27 Topallama, 27-36 Düşme, 36-45 Oturma, 45-51 Bastonla yürüme ve 51-55 Değnekle yürüme) ve (b) Sınıf içi benzerliklerin kıyaslanması (x eksenini boy iterasyonlarını içermektedir: 1-114 ilk, 114-228 ikinci, 228-342 üçüncü, 342-456 dördüncü, and 456-570 beşinci).

2.3 Mikro-Doppler İmzaları

Bu çalışmada, mikro-Doppler imzalarının zaman-frekans gösterimini oluşturmak için kısa zamanlı Fourier dönüşümü (STFT) kullanılmıştır. STFT şu şekilde tanımlanır:

$$STFT(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \quad (2.5)$$

Burada $x[n]$ alınan sinyali ve $w[m]$ pencere fonksiyonunu temsil etmektedir. Bu tezde, 2048 örnek genişliğinde bir Hamming fonksiyonu pencereleme işlemi için kullanılmıştır. Pencere 128 örneklilik çakışmalar ile kaydırılarak zaman sinyali taranmış ve her Fourier dönüşümü 4096 Hızlı Fourier Dönüşümü (FFT) bin'i içererek hesaplanmıştır. Her spektrogram, maksimum 4 saniyeden oluşacak şekilde kesilmiş, gri skalaya dönüştürülmüş, normalize edilmiş ve 90x120 çözünürlüğe aşağı örneklenecek şekilde olarak kaydedilmiştir.

Son olarak, gürültü altında modellerin sunduğu başarımın incelenmesi için aşağıdaki adımlar ile spektrogramlara gauss gürültüsü eklenmiştir:

$$S_n = S + \sigma \text{randn}(M, N) \quad (2.6)$$

Burada S_n gürültü eklenmiş spektrogramı, S orjinal spektrogramı, M frekanstaki örnek sayısını ve N ise zamandaki örnek sayısını belirtmektedir. İstenilen SNR değeri için varyans σ , gövde yanıtının gücü olan a üzerinden tanımlanmakta, bu da spektrogramdaki tepe noktalar üzerinden şu şekilde hesaplanmaktadır:

$$\sigma^2 = a/10^{SNR/10} \quad (2.7)$$



3. GELENEKSEL YÖNTEM

Literatürde, mikro-Doppler imzalarının sınıflandırılması için en sık kullanılan yöntemlerden biri, ön-tanımlı özniteliklerin ham veriden veya spektrogram üzerinden çıkartılması ile bir sınıflandırıcı eğitmektir.

3.1 Öznitelik Tanımları

Literatürde fiziksel, dönüşüm tabanlı ve konuşma işleme tabanlı oluşturulmuş öznitelikleri içeren bir çok çalışma mevcuttur [?]. Tez kapsamında, literatürde önerilmiş özniteliklerin büyük bir kısmı kullanılmış olup bu bölümde detaylı bir şekilde anlatılacaktır.

3.1.1 Fiziksel öznitelikler

Fiziksel öznitelikler, spektrogram veya cadence hız diyagramı (CVD) üzerinden insan hareketiyle doğrudan ilintili olanların ölçümü ile elde edilir. CVD basitçe spektrogramın her bir frekans bini için Fourier dönüşümü alınarak aşağıdaki gibi hesaplanır [?] [?] [?]:

$$\Delta(v, \omega) = \sum_{n=0}^{N-1} |STFT(n, \omega) e^{-\frac{j2\pi n v}{N}}| \quad (3.1)$$

Sezgisel olarak CVD spektrogram üzerinde farklı hızların ne sıklıkla tekrar ettiğinin bir ölçütüdür. Bu çalışmada kullanılan fiziksel öznitelikler şu şekildedir: Gövde yanıtının bant genişliği, genel doppler bant genişliği, ortalama gövde yanıtı, Üst zarfın minimum, maksimum ve ortalama değeri, alt zarfın minimum, maksimum ve ortalama değeri, üst ve alt zarf ortalamaları arasındaki fark, CVD'nin temel frekansı, ikinci ve üçüncü harmonikleri.

Fiziksel özniteliklerden gövde yanıtı, spektrogram üzerinden tüm frekans bin'leri için enerjinin en yüksek olduğu yerler olarak hesaplanır. Bu bin'ler içinden maksimum ve minimum frekans değerleri arasındaki fark gövde yanıtının bant genişliğini vermektedir. Toplam doppler bant genişliği, spektrogramın maksimum ve minimum frekans değerleri arasındaki fark olarak bulunur. Üst ve alt zarf ise bir eşikleme yardımı ile bulunur.

3.1.2 Dönüşüm tabanlı öznitelikler

Dönüşüm tabanlı özniteliklere örnek olarak ayrık kosinüs dönüşümü (DCT) verilebilir. Bu çalışmada ayrık kosinüs dönüşümünün ilk 10 bileşeni kullanılmıştır. Mikro-Doppler imzası $f_d(t)$ olan bir sinyalin DCT bileşenleri aşağıdaki gibi hesaplanır:

$$C(k) = h(k) \sum_{t=0}^{T-1} f_d(t) \cos\left[\pi\left(t + \frac{1}{2}\right)\frac{k}{T}\right] \quad (3.2)$$

burada T gözlemlenen sinyalin uzunluğunu temsil eder, $k \in [0, t-1]$ ve $h(k)$ aşağıdaki gibi tanımlanır:

$$h(k) = \begin{cases} \sqrt{\frac{1}{T}} & \text{icin } k = 0 \\ \sqrt{\frac{2}{T}} & \text{diğer} \end{cases} \quad (3.3)$$

3.1.3 Ses işleme tabanlı öznitelikler

Ses işleme öznitelikleri olarak, ilk 3 kepsstral katsayı ile birlikte 101 doğrusal öngörülü kodlama katsayısı (LPC) kullanılmıştır. LPC katsayıları doğrudan yansıma sinyali üzerinden hesaplanmıştır. LPC hesaplanırken yapılan işlem $x[n]$ anındaki sinyali geçmiş değerlerin doğrusal kombinasyonları şeklinde temsili üzerinedir ve şöyle hesaplanır:

$$\hat{x} = \sum_{k=1}^p a[k]x[n-k] \quad (3.4)$$

burada $a[k]$ değerleri LPC'leri ve p değeri ise toplam LPC sayısını belirtir. LPC'leri hesaplayabilmek için Denklem (3.4)'te belirtilen model ile gerçek sinyal arasındaki fark, $e[n] = x[n] - \hat{x}[n]$, minimize edilmelidir. Bu işlemi birçok yöntem ile yapmak mümkündür. Tez kapsamında bu problem otokorelasyon sonrası Levinson-Durbin özyinelemesi ile çözülmüştür.

Bir diğer ses işleme tekniği olan kepsstrum, $c[n]$, yansıma sinyalinin $x[n]$ Ayrık Fourier Dönüşümü'nün(DFT) log büyüklüğünün ters DFT'si olarak tanımlanır:

$$c[n] = F^{-1}[\log|F[x[n]]|] \quad (3.5)$$

burada $F[.]$ Fourier dönüşümünü simgelemektedir.

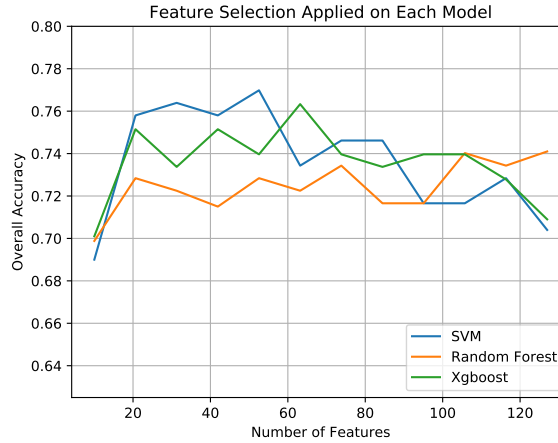
Özetlemek gerekirse, 10 DCT, 3 kepsstral, 13 fiziksel ve 101 LPC olmak üzere toplamda 127 öznitelik çıkartılmıştır.

3.2 Öznitelik Seçimi ve Sınıflandırma Kıyaslaması

Çokboyutluluğun laneti nedeniyle, her durumda olası tüm özniteliklerin kullanılması en iyi performansı garanti etmemektedir. Literatüre bakıldığında öznitelik seçiminin sınıflandırma başarımını önemli ölçüde arttırdığı görülmüştür [?] [?] [?]. Bu tez kapsamında öznitelik seçimi yöntemi olarak Sıralı Geri Eleme (SBE) yöntemi kullanılmıştır [?]. SBE, belirli bir sınıflandırıcı için en iyi başarıyı veren özniteliklerin kombinasyonunu bulmak için açgözlü (greedy) bir şekilde arama yapan bir sarıcı (wrapper) yöntemidir. SBE tüm öznitelikleri kullanarak işleme başlar ve ardından öznitelikleri iteratif olarak öznitelik uzayından çıkartmaya başlar. Bu prosedür, belirlenen sayıda öznitelik seçilinceye kadar tekrarlanır.

Bu çalışmada, literatürde yaygın bir şekilde kullanılan 3 farklı sınıflandırıcı ele alınmıştır: Destek Vektör Makinası (SVM), Rastgele Orman Sınıflandırıcısı (RF) [?] [?] [?] ve Ekstrem Gradyan Yükselticisi (Xgboost) [?]. İlk olarak tüm sınıflandırıcılar için optimal hiperparametreler ızgara araması (grid search) yöntemi ile bulunmuştur. SVM için doğrusal, polinom ve radyal taban fonksiyonu (RBF) çekirdek fonksiyonları denenmiştir ve doğrusal çekirdek yönteminin en yüksek başarıyı verdiği görülmüştür. Xgboost ve RF sınıflandırıcıları için model hiperparametreleri iki parametreden oluşmaktadır: ormandaki ağaç sayısı ve ağacın maksimum derinliği. ızgara araması sonucu, RF için en iyi sonuç, 50 ağaç ve 20 derinlik ile, Xgboost için ise 50 ağaç ve 10 derinlik ile elde edilmiştir.

Sınıflandırıcı parametrelerinin optimizasyonundan sonra, sınıflandırma başarımı, Şekil 3.1'de gösterildiği gibi öznitelik sayısının bir fonksiyonu olarak karşılaştırılmıştır.



Şekil 3.1: SVM, RF ve Xgboost yöntemleri için öznitelik sayısına bağlı sınıflandırma sonuçları.

En iyi performans, SBE ile seçilen 50 öznitelik ile eğitilen doğrusal çekirdekli (kernel) SVM ile elde edilmiştir. SVM için SBE ile seçilen öznitelikler şunlardır: ortalama gövde frekansı, üst zarfın ortalaması, alt zarfın ortalaması, ilk 2 CVD özniteliği, ilk 2 cepstral katsayı, 37 LPC katsayısı ve 5 DCT özniteliği. Şekil 3.1 üzerinden öznitelik seçiminin etkisinin, RF ve Xgboost sınıflandırıcılarında biraz daha az belirgin olduğu görülmektedir. Bunun sebebi olarak ağaç tabanlı modellerin eğitim esnasında saf olmama (impurity) tabanlı bir öznitelik sıralaması yöntemi kullanması gösterilebilir.

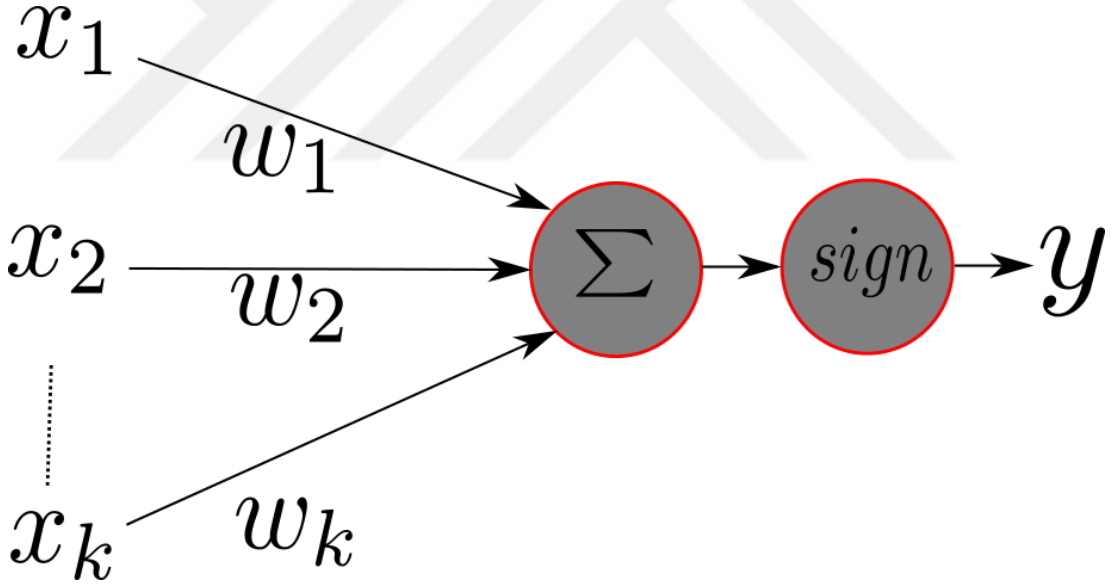


4. YAPAY SİNİR AĞLARI

Tezin bu kısmında Yapay Sinir Ağları(NN)'nin gelişimi, güncel teknoloji ve tez kapsamında kullanılmış DNN mimarilerinin detaylı anlatımı verilecektir.

4.1 Yapay Sinir Ağlarına Giriş

Yapay Sinir Ağları (NN) ilk olarak 1943 yılında McCulloch ve Pitts tarafından önerilmiştir [?]. Şekil 4.1 üzerinden genel mimarisi görülen McCulloch Pitts nöronu oldukça basit bir yapıdır ve temelde yaptığı işlem x_1, x_2, \dots, x_k girdilerini, w_1, w_2, \dots, w_k ağ ağırlıklarıyla çarpıp topladıktan sonra bir aktivasyon fonksiyonu olan signum fonksiyonundan geçirip ikili (binary) değerler üretmektedir. Yani özünde ikili bir sınıflandırıcıdır. McCulloch ve Pitts tarafından bu yapının mantıksal operasyonlar için kullanılabileceği gösterilmiş, ancak bu model için herhangi bir öğrenme algoritması önerilmemiştir.



Şekil 4.1: McCulloch Pitts Nöronu'nun genel yapısı.

Sonrasında 1958 yılında Frank Rosenblatt tarafından Perceptron algoritması geliştirilmiştir [?]. Önerilen Perceptron yapısı McCulloch-Pitts nöronuna oldukça benzeyen ikili bir sınıflandırıcıdır ve aktivasyon fonksiyonu olarak McCulloch Pitts modelindeki signum basamak fonksiyonu kullanılabileceği gibi doğrusal olmayan alt uzay öznitelik temsilleri elde etmek için doğrusal olmayan sigmoid ve hiperbolik tanjant aktivasyonları da kullanılabilir. Sigmoid ve Hiperbolik Tanjant aktivasyonları Şekil 4.2 üzerinden görülebilir. [?]’in asıl katkısı, Perceptron için bir öğrenme algoritmasının da önerilmiş olmasıdır.

$$\sigma(\Sigma) = \frac{1}{1 + e^{-\Sigma}}$$

$$\tanh(\Sigma) = \frac{e^{\Sigma} - e^{-\Sigma}}{e^{\Sigma} + e^{-\Sigma}}$$



Şekil 4.2: Sigmoid ve Hiperbolik Tanjant aktivasyonları.

Perceptron öğrenme algoritmasında öncelikle genellikle -1 ile 1 aralığında olacak şekilde rastgele sayılar ile ağırlıklar (w değerleri) başlatılır. Sonrasında:

$$d(n) = \begin{cases} +1 & \text{eger } x(n) \in A \text{ sınıfı} \\ -1 & \text{eger } x(n) \in B \text{ sınıfı} \end{cases} \quad (4.1)$$

Veri etiketlerine bağlı olarak herhangi bir girdi için Denklem (4.1)'in yanlış olduğu durumlarda, yani yanlış sınıflandırılan örnekler için ağırlık vektörü w değeri aşağıdaki ifadeye göre güncellenir:

$$w_i = w_i + \eta d(n) x_i(x) \quad (4.2)$$

Burada η öğrenme oranını, x_i ise girdiyi ifade etmektedir. Bu işlem tüm girdiler doğru sınıflandırılana kadar tekrar eder. Bu öğrenme algoritması ile Perceptron modelinin, rastgele başlatılan ağırlık değerleri ile öğrenme işlemi otomatik olarak gerçekleştirilmekte olduğu gösterilmiştir.

Rosenblatt'ın iddiası şudur: Çözülebilir ağırlıkların olduğu her problem için Perceptron optimal sonuçlara yakınsar. Ancak bu düşünce 1969 yılında Marvin ve Papert tarafından xor örneği üzerinden çürütülmüştür [?]. Makalede tek katmanlı Perceptron modelinin, yalnızca doğrusal ayrışabilen problemlere çözüm getirebildiği ve basitliğine rağmen doğrusal ayrışamayan xor problemine çözüm sağlayamadığı açıkça gösterilmiştir. Sonrasında Hopfield tarafından Perceptron yapısının çok katmanlı hale getirilmesi ile xor probleminin çözülebildiği gösterilmiştir.

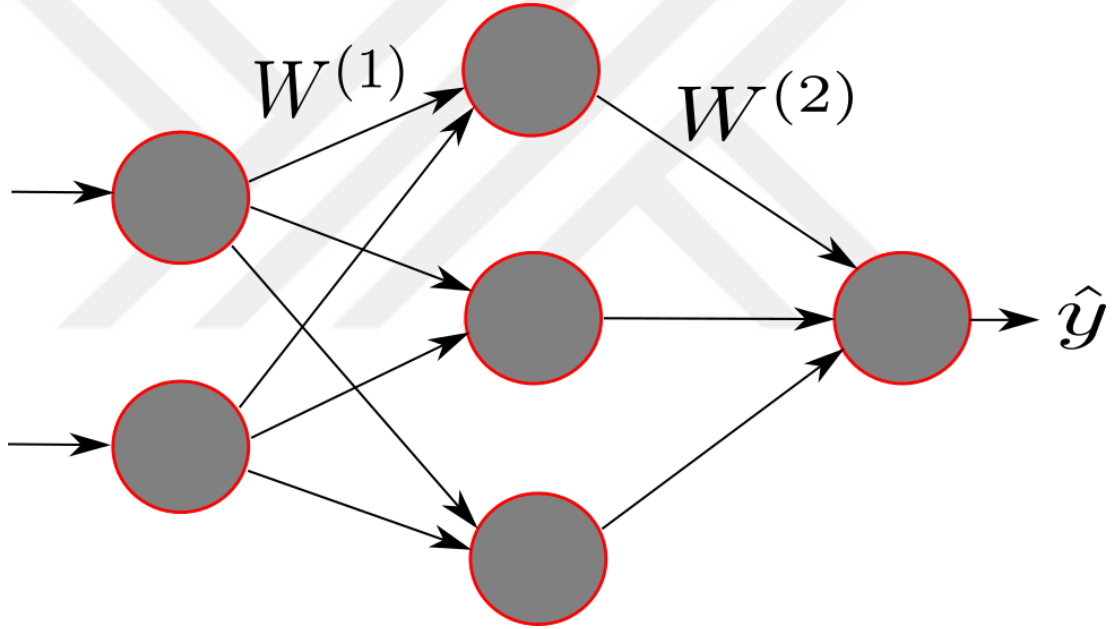
1986 yılında Perceptronların eğitimi için geri yayılım algoritması önerilmiştir [?]. Geri yayılım algoritması, Perceptron öğrenme kuralındaki gibi hatayı doğrudan aktarmak yerine, hatanın ağırlıklara göre kısmi türevinin geri yansıtılarak ağırlıkların öğrenilmesini esas alır. Bu algoritma güncel öğrenme algoritmaları için de bir temel oluşturduğu için detaylı anlatımı Bölüm 4.2'de verilmiştir.

Sonrasında 1989 yılında Lecun tarafından Konvolüsyonel Yapay Sinir Ağları (CNN) önerilmiştir [?]. Görü sisteminden ilham alan bir yapı olan CNN hiyerarşik öznelik öğrenimini esas almaktadır. Bu yapı 1970'lerde önerilen Neocognitron [?] yapısının devamıdır ve güncel derin öğrenme literatürü için büyük önem arz etmektedir. Sonrasında 2006 yılında Hinton Gözetimsiz Ön-eğitim algoritmasını önermiştir [?]. Bu

çalışma Derin Öğrenme literatürünün başlamasına sebebiyet veren çalışma olarak gösterilmektedir. Bu iki algoritma detaylı olarak Bölüm 4'te anlatılacaktır.

4.2 Geri Yayılım Algoritması

Yapay Sinir Ağlarının temelde yaptığı iki genel işlem vardır: İleri Besleme ve Geri Yayılım. İleri besleme fazında, verilen girdiler son katmana kadar çeşitli işlemlerden geçerek ilerlerler ve bir çıktı oluştururlar. Geri yayılım fazında ise, bu çıktının ürettiği hata, ağı son katmanındaki ağırlıklardan ilk katmanındaki ağırlıklara kadar geri yansıtılarak ağı ağırlıkları güncellenir/eğitilir. Bu hata fonksiyonu için basitçe, regresyon problemleri için ortalama kare hatası, sınıflandırma problemleri için ise negatif log benzerliği örnek verilebilir. Tez kapsamında geri yayılım algoritması, sezgisel olarak anlaşılması daha basit olduğu düşünüldüğü için bir regresyon problemi üzerinden anlatılacaktır. Anlatım için model mimarisi olarak Şekil 4.3 üzerinde gösterilen ve aktivasyon fonksiyonu olarak sigmoid kullanan model ele alınacaktır.



Şekil 4.3: Bir ileri beslemeli tam bağlı yapay sinir ağı.

Bu model girdi katmanı, çıktı katmanı ve gizli katmandan oluşan sıg bir NN'dir. Modele girdi olarak 2 boyutlu X vektörünü verilirse, ileri besleme modunda bu vektör ilk olarak bir vektör olan $W^{(1)}$ ağırlıkları ile çarpılır:

$$z^{(2)} = XW^{(1)} \quad (4.3)$$

Burada notasyon olarak $W^{(1)}$ 1. katmandaki ağırlıkları temsil etmektedir. Daha sonra elde edilen çıktı Şekil 4.2'de gösterilen sigmoid aktivasyon fonksiyondan geçirilirse:

$$\sigma^{(2)} = f(z^{(2)}) \quad (4.4)$$

Burada $f(t) = \frac{1}{(1+e^{-t})}$ olarak tanımlanmaktadır. Sonrasında bu çıktı gizli katman ile çıktı katmanı arasındaki ağırlık değerleri, $W^{(2)}$ ile çarpılır.

$$z^{(3)} = \sigma^{(2)}W^{(2)} \quad (4.5)$$

Bu sonuç da yine aynı şekilde aktivasyon fonksiyonuna sokulur ve çıktı \hat{y} elde edilir.

$$\hat{y} = f(z^{(3)}) \quad (4.6)$$

Ağ ilk başlatıldığında W değerleri rastgele bir şekilde başlatılır. Bu değerleri optimize etmek için aşağıdaki masraf fonksiyonu minimize edilmelidir:

$$J(W) = \sum_{x=1}^N \frac{1}{2} (y - \hat{y})^2 \quad (4.7)$$

Burada N toplam girdi sayısını, y her bir X girdisine karşılık gelen sürekli değeri, J ise masraf fonksiyonunu ifade etmektedir. Burada masraf fonksiyonunu Denklem (4.3 - 4.6) cinsinden tek bir büyük denklemde toplarsa:

$$J(W) = \sum_{x=1}^N \frac{1}{2} (y - f(f(XW^{(1)})W^{(2)}))^2 \quad (4.8)$$

ifadesi elde edilir. Bu ifadenin minimize edilmesi için gradyan inişi algoritması kullanılabilir. Burada amaç, iteratif olarak W değerlerine bağlı hatanın gradyanını hesaplayıp, gradyanın tersi istikamette ilerlemektir.

Denklem (4.8)'in minimize edilebilmesi için $\frac{\partial J}{\partial W^{(1)}}$ ve $\frac{\partial J}{\partial W^{(2)}}$ ayrı ayrı hesaplanabilir. Burada temel kalkülüs kuralı olan zincir kuralından faydalanılacaktır. İlk olarak $\frac{\partial J}{\partial W^{(2)}}$ hesaplanır:

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial \frac{1}{2} (y - \hat{y})^2}{\partial W^{(2)}} = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial W^{(2)}} \quad (4.9)$$

Buradan Denklem (4.5) ve Denklem (4.6) hatırlanırsa bu üstteki ifade şu ifadeye dönüşür:

$$\frac{\partial J}{\partial W^{(2)}} = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial W^{(2)}} \quad (4.10)$$

Burada $\frac{\partial \hat{y}}{\partial z^{(3)}}$ ifadesi sigmoidin türevinden $\frac{e^{-z}}{(1+e^{-z})^2}$ olarak ifade edilebilir. $\frac{\partial z^{(3)}}{\partial W^{(2)}}$ ise zincir kuralından $\sigma^{(2)}$ 'dir.

Aynı şekilde $\frac{\partial J}{\partial W^{(1)}}$ zincir kuralı ile hesaplandığında şu ifade elde edilir:

$$\frac{\partial J}{\partial W^{(1)}} = -(y - \hat{y}) X^T f'(z^{(3)}) (W^{(2)})^T f'(z^{(2)}) \quad (4.11)$$

Burada f sigmoid fonksiyonunu X girdi matrisini T ise transpoz operasyonunu ifade etmektedir. Gradyan değerleri zincir kuralı ile hesaplandıktan sonra $W^{(i)} = W^{(i)} -$

$\sigma \frac{\partial J}{\partial W^{(i)}}$ ifadesine göre W değerleri güncellenir. Geri yayılım algoritması ile ağı eğitilmesi basitçe bu örnek üzerinden özetlenebilir.

4.3 Derin Öğrenme

Derin öğrenme, modern GPU'ların artan paralel işlem kabiliyetleri ve algoritmik ilerlemeler sebebiyle yakın zamanda ortaya çıkmış bir makine öğrenmesi yöntemidir. Aslında temel olarak NN'lerden farkı, mimarinin 2'den fazla gizli katmandan oluşmasıdır, yani yöntemsel olarak geçmiş NN araştırmalarından ilham almaktadır. Burada DNN'lerin neden geçmişten beridir popüler olmadığı konusu üzerine durulmalıdır. Bunun 3 temel sebebi vardır: İşlem gücü yetersizliği, veri eksikliği ve kaybolan gradyan problemi. Derin modellerde tipik olan milyonlarca ağırlığın öğrenilmesi için, paralel işlem kabiliyetleri sınırlı olan CPU'lar geride bırakılmış ve GPU'lara yönelinmiştir. Derin modeller evrensel yakınsayıcılar olduklarından, entropik kapasiteleri çok yüksektir, dolayısıyla aşırı öğrenmeye yatkındırlar. Problemin karmaşıklığına bağlı olarak iyi bir genelleme öğrenebilmeleri için fazla miktarda veri gerekmektedir. Veri günümüzde eskiye göre çok daha yaygın bir şekilde bulunabildiğinden bu da artık bir problem olmaktan çıkmıştır. Kaybolan gradyan problemi ise basitçe, hatanın türevinin geri yayılım ile aktarılması durumunda sigmoid tabanlı sıkıştırıcı aktivasyon fonksiyonları kullanıldığında görülmektedir [?]. Sigmoid tabanlı aktivasyon fonksiyonları girdileri sıkıştırdıkları için, geri yayılım esnasında ilk katmanlara ulaşan hata 0 ile 1 arasındaki birçok değer için çarpılması sonucu çok küçülmekte ve ilk katmanlardaki ağırlıkların eğitilememesine sebebiyet vermektedir. Son yıllarda bu problemin çözümü için çeşitli algoritmik yenilikler önerilmiştir. Bunlardan en basiti Doğrusal Doğrusal Ünite (Relu) aktivasyonları kullanmaktır [?]. Matematiksel olarak Relu şu formdadır;

$$f(x) = \max(0, x) \quad (4.12)$$

Relu aktivasyon değerini sıkıştırmadığı için kaybolan gradyan problemine sebebiyet vermez ve genel olarak doğrusal gibi görünse de süperpozisyon ilkesini sağlamadığı için doğrusal olmayan bir aktivasyon sağlamaktadır. Ayrıca Relu'nun bir diğer avantajı olarak nöronların rastgele başlatıldığı durumda çoğu nöron sıfırın altında değer üreteceği için (hatta -1 ile 1 arasında tekdüze dağılım ile başlatılan nöron ağırlıkları için nöronların yalnızca yarısı aktive olacaktır) seyrek bir ağ yapısı, dolayısıyla seyrek bir veri temsili sağlar [?]. Bu sebeplerden ötürü, çeşitli varyantları hariç, Relu günümüzde DNN mimarilerinde sıkça kullanılmaktadır ve bu tez kapsamında da tüm mimariler için aktivasyon fonksiyonu olarak Relu kullanılmıştır. Buna ek olarak kaybolan gradyan problemini çözen bir diğer yöntem ağ ağırlıklarının rastgele başlatılması yerine, katmandan katmana gözetimsiz bir ön-eğitim yöntemi kullanmaktır [?].

4.4 Derin Öğrenme Mimarileri

Bu başlık altında tez kapsamında kullanılan Derin Öğrenme mimarileri ve ağ eğitim teknikleri anlatılacaktır.

4.4.1 Otokodlayıcı

Temel olarak girdi, çıktı ve gizli katmanlardan oluşan Otokodlayıcı(AE) yapısı, verilen girdiyi, çeşitli kısıtlar altında çıktı olarak üretmeye çalışan bir ileri beslemeli yapay sinir ağıdır. Başka bir deyişle, verilen girdi vektörü x için AE $h_w(x) \approx x$ yakınsamasını sağlamaya çalışır. 2006 yılında ağ ağırlıklarının başlatılması için gözetimsiz ön-eğitim algoritmasının [?] önerilmesiyle ortaya çıkan AE yapılarının, özellikle az sayıda etiketli veri mevcut olduğunda oldukça etkili olduğu görülmüştür [?] [?]. Gözetimsiz ön-eğitim basitçe birbirine bağlı bir kodlayıcı ve bir kod çözücü ağ tarafından uygulanır.

Verilen bir girdi vektörü X için kodlayıcı, girdinin doğrusal olmayan haritalamasını şu şekilde hesaplar

$$E = \sigma(WX + b). \quad (4.13)$$

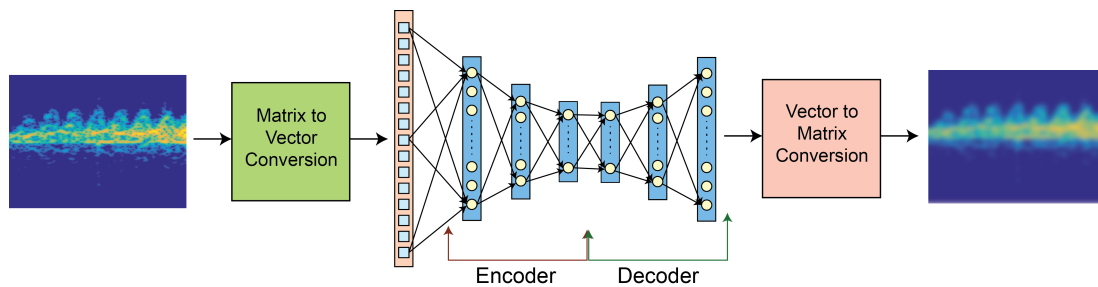
Burada, σ doğrusal olmayan aktivasyon fonksiyonunu, W ağ ağırlıklarını ve b sabit terimi temsil etmektedir. Kodlayıcının öğrenmiş olduğu öznelikler daha sonrasında bir kod çözücü tarafından girdi X 'i tekrar oluşturmak için aşağıdaki işlemde geçer

$$Z = \sigma(\tilde{W}E + \tilde{b}). \quad (4.14)$$

Burada \tilde{W} ve \tilde{b} kod çözücünün ağırlık ve sabit terimini temsil eder. Gözetimsiz ön-eğitim'de ağ ağırlık ve sabit terimini $\theta = [W, b, \tilde{W}, \tilde{b}]$ ayarlayarak her bir x_i değeri için aşağıdaki masraf fonksiyonunu minimize etmeye çalışır

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N (x_i - z_i)^2 \quad (4.15)$$

Burada ağın doğrudan birim matrisi öğrenmesini engellemek adına Denklem (4.15)'e ve ağın kendisine çeşitli kısıtlar eklenmelidir. Bunlardan ilki, gizli katmandaki nöron sayısını girdi olarak verilen vektörün boyutluluğundan az tutmaktır. Bu durum, ağı, düşük boyutlu bir uzayda doğrusal olmayan işlemler ile üst uzaydaki girdiyi tekrar oluşturmaya zorlayacaktır. Buna ek olarak Denklem (4.15)'e bir seyreklik parametresi eklenmelidir. Bu parametre, ağı verilen girdi vektörleri arasındaki korelasyonları öğrenmeye zorlar [?]. Seyreklik parametresi eklendikten sonra Denklem (4.15) aşağıdaki hale gelir



Şekil 4.4: 3 Katmanlı AE yapısı, kodlayıcı katmanlarında sırasıyla 200-100-50 nöron, kod çözücü ise 50-100-200 nöron içermektedir.

$$\operatorname{argmin}_{\theta} J(\theta) = \frac{1}{N} \sum_{i=1}^N (x_i - z_i)^2 + \beta \sum_{j=1}^h KL(p||p_j). \quad (4.16)$$

Burada h gizli katmandaki nöron sayısını, β seyreklik oranını ve $\sum_{j=1}^h KL(p||p_j)$ ise p ve p_j ortalamalı Bernoulli rastgele değişkenleri arasındaki Kullback-Leibler (KL) ıraksamasını belirtmektedir. İki rastgele değişken arasındaki KL ıraksaması ise aşağıdaki gibi verilmektedir

$$KL(p||p_j) = p \log\left(\frac{p}{p_j}\right) + (1-p) \log\left(\frac{1-p}{1-p_j}\right), \quad (4.17)$$

Burada p_j , gizli katmandaki j . nöronun aktivasyonunu, p ise arzulanan aktivasyon değerini belirtmektedir. KL ıraksama terimi basitçe gizli nöron aktivasyonlarının belirli bir p aralığında kalmasını zorlar.

Gözetimsiz ön-eğitim bittikten ve dolayısıyla ağırlıkların kendisinden gelen bilgi ile başlatıldıktan sonra, kod çözücü kısım ağırlıklardan çıkarılır ve geriye kalan kodlayıcı kısım gözetimli olarak sınıflandırma işlemi için kullanılır. Bunun için kodlayıcı kısmın sonuna arzulanan sınıf sayısı kadar nöron içeren bir softmax sınıflandırıcısı eklenir. Softmax sınıflandırıcısı, lojistik regresyon modelinin çok-terimli (multinomial) versiyonudur. Verilen bir girdi x_i için, softmax fonksiyonu, bu girdinin her bir sınıf değeri $k = 1, 2, \dots, K$ 'ya ait olma olasılığı $P(y_k|x_i)$ 'i hesaplar. Başka bir deyişle, girdi x_i 'nin, sınıf etiketi y_k olma olasılığı kestirilir. Matematiksel olarak sınıf olasılığı p_k şu şekilde verilir:

$$p(y = k|x_i) = \frac{e^{\theta_k x_i}}{\sum_{k=1}^K e^{\theta_k x_i}}. \quad (4.18)$$

Ağın ağırlık ve sabit terimleri θ aşağıdaki masraf fonksiyonu minimize edilerek çözülür

$$J(\theta) = - \sum_{i=1}^N \sum_{k=1}^K 1\{y^i = k\} \log \frac{e^{\theta_k x_i}}{\sum_{k=1}^K e^{\theta_k x_i}}, \quad (4.19)$$

Burada $1\{\cdot\}$ indikatör fonksiyonunu temsil eder. İndikatör fonksiyonu içindeki eşitlik sağlanıyorsa 1 değeri, sağlanmıyorsa 0 değeri üretir. N etiketlenmiş veri sayısını temsil eder. Denklem (4.19) gradyan tabanlı bir algoritma ile çözülür. Bu gözetimli eğitim işlemine literatürde hassas/ince ayar (fine tuning) denir. İnce ayar işlemi yapılırken gözetimsiz ön-eğitim'den farklı olarak ortalama kare hatası değil, capraz entropi hatasının kullanıldığına dikkat edilmelidir.

AE yapıları alt katmanların çıktılarının üst katmanlara girdi olarak verildiği bir hiyerarşi ile derin olarak kurulabilir. Bu tez kapsamında kullanılan AE yapısının masraf fonksiyonları Adaptif Moment Tahminlemesi (ADAM) [?] ile hesaplanmıştır. Kullanılan AE yapısı ise Şekil 4.4 üzerinden görülebilir.

4.4.2 Konvolüsyonel Yapay Sinir Ağları

Konvolüsyonel Sinir Ağları (CNN) literatüre 1990'larda girmiş olan bir yapı olduğundan, bu yapıyı doğrudan derin öğrenme adı altında incelemek doğru olmasa da son yıllarda derin öğrenmenin yarattığı en büyük etki CNN'lerden gelmiştir denebilir. CNN'ler lokalde birbirine bağlı öznelikleri öğrenme konusundaki başarıları sayesinde görüntü sınıflandırma konusunda yakın zamanda çığır açmışlardır. Bir CNN genel olarak üç bileşenden oluşmaktadır: Konvolüsyonel katman, havuzlama katmanı, ve tam bağlı katman [?]. Konvolüsyonel katmanda filtreler girdi matrisinin alıcı alanı ile bir kayan pencere işlemi ile konvolüsyona sokulurlar ve lokalda bağlı öznelikleri öğrenirler. İlk katmanlarda kenar, köşe ve çizgi gibi basit öznelıklar öğrenilirken katmanlar derinleştikle daha soyut öznelıklar öğrenilir. Matematiksel olarak verilen girdi matrisi P için CNN'deki m . nöron şunu hesaplar:

$$M[i, j] = \sigma \left(\sum_{x=-2k-1}^{2k+1} \sum_{y=-2k-1}^{2k+1} f_m[x, y] P[i-x, j-y] + b \right), \quad (4.20)$$

burada orijin matrisin merkezi olarak tanımlanmıştır ve matrisin köşesi orijinden x ve y yönlerinde k eleman uzaktadır. Konvolüsyonel filtrenin bir kaydırma miktarı $2k + 1$ kadardır, M aktivasyon haritasını temsil etmektedir, f_m ise m . konvolüsyonel filtreyi temsil eder ve σ aktivasyon fonksiyonudur.

Genel olarak konvolüsyonel katmandan sonra bir maksimum havuzlama katmanı kullanılır. Bu katman basitçe bir aşağı-örnekleme prosedürüdür ve genel olarak 2×2 'lik komşulukların maksimum değeri alınır. Havuzlama hem ileri katmanların işlem yükünü azaltır hem de ağa bir miktar 'kaydırma değişmezliği (translation invariance)' katar.

Tam bağlı katmanlar ise genelde ağın en sonuna bağlanır. Bu katmanda, konvolüsyonel katmanlarda öğrenilen öznelıkların doğrusal olmayan kombinasyonları öğrenilir. Literatürde yaygın olarak kullanılan bir regülarizasyon algoritması olan bırakma (Dropout) kullanılır [?]. Bu algoritma her iterasyonda basitçe rastgele seçilmiş nöronları ileri besleme ve geri besleme fazlarında inaktif yapar ve ağı farklı doğrusal olmayan öznelik kombinasyonlarını bulmaya zorlar.

CNN için konvolüsyonel katmanlarda zaman karmaşıklığı [?] şu şekilde gösterilmektedir:

$$\mathcal{O} \left(\sum_{l=1}^d n_{l-1} \cdot s_l^2 \cdot n_l \cdot m_l^2 \right) \quad (4.21)$$

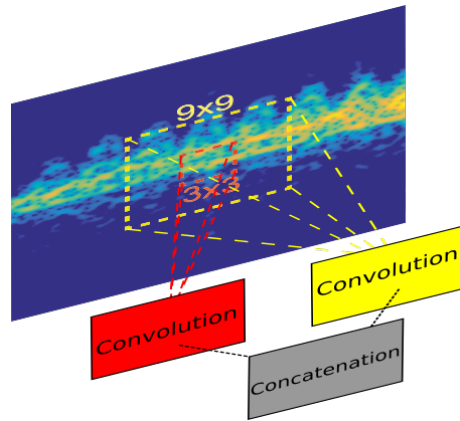
burada l konvolüsyonel katman indeksini, d derinliği, n_{l-1} l . katmana verilen girdi sayısını, s_l filtrenin uzamsal boyutunu ve son olarak m_l öznelik çıktısının boyutunu belirtir. Filtrenin uzamsal boyutu ve öznelik çıktısının boyutunu zaman kompleksitesini üstel bir şekilde arttırmaktadır. Derinlik ve girdi boyutluluğu ise doğrusal bir artışa sebep olmaktadır. Bununla birlikte derinliğin artırılması, performansa diğer DNN parametrelerinden çok daha fazla etki eder. Özellikle ileri beslemeli tam bağlı yapay sinir ağları için teorik olarak derinliği sadece 1 katman arttırmanın, o katmanda kullanılan nöron sayısını arttırmaya göre üstel oranda fayda sağladığı gösterilmiştir [?]. [?]'de CNN'ler için de benzer sonuçlar elde edilmiştir.

Bu tez kapsamında [?]’ye benzer bir filtre bağlama tekniği kullanılmıştır. Bu teknik birbirinden farklı boyutlarda konvolüsyonel filtrelerin kullanılmasını sağlar. Bu tez kapsamında her katmanda iki farklı boyutta filtre birbirine bağlanmıştır. Büyük filtreler daha genel öznitelikleri öğrenirken, küçük filtreler ince detayları öğrenmeye odaklanırlar. Bu yöntem, hesaplama masrafını arttırsa da, sınıflandırma başarımını da önemli ölçüde arttırmaktadır. Şekil 4.5 üzerinden filtre bağlama tekniğinin genel yapısı görülebilir.

Çizelge 5.2’de filtre boyutunun, katman genişliğinin ve katman sayısının sınıflandırma başarımına etkisi gösterilmiştir. Bu analiz kapsamında 3 katmanlı ve her katmanında 30 filtre kullanan bir CNN yapısının kullanılması ve filtre boyutları olarak 3x3 ve 9x9’lük filtrelerin seçilmesi en yüksek başarıyı vermektedir. Her konvolüsyonel katmandan sonra 2x2’lik maksimum havuzlama uygulanmıştır. Ağın sonunda öğrenilmiş öznitelikler vektörize edilmiş ve tam bağlı katmana girdi olarak verilmiştir. 2 tam bağlı katman kullanılmış olup her bir katmanda 150 nöron kullanılmıştır. Her tam bağlı katmanda 0.5 ihtimalle bırakma kullanılmıştır. Son olarak ağ bir softmax sınıflandırıcısına bağlanmıştır ve softmax toplam sınıf sayısı kadar nöron içermektedir. Denklem (4.19) ADAM ile optimize edilmiştir. Tez kapsamında kullanılan genel CNN mimarisi Şekil 4.6 üzerinden görülebilir.

4.4.3 Konvolüsyonel Otokodlayıcı

Konvolüsyonel Otokodlayıcılar (CAE) CNN’lerin lokalde bağlı filtreleme yeteneği ile AE’lerin gözetimsiz ön-egitim yeteneklerini birleştirirler [?]. AE topolojisinden farklı olarak, CAE’ler kodlayıcı kısmında konvolüsyonel filtreler, kod çözücü kısımda ise dekonvolüsyonel filtreler içermektedir. Dekonvolüsyonel filtreler konvolüsyonel filtrelerin transpozlanmış halleri olabileceği gibi, bu tez kapsamında yapıldığı gibi sıfırdan da öğrenilebilmektedirler. Buna ek olarak, CAE mimarisinde her dekonvolüsyonel katmanı bir tersine-havuzlama işlemi takip eder [?]. Tersine-havuzlama işlemi havuzlama işlemi esnasında maksimum değerlerin konumlarının kayıt altına alınıp, bu indislerdeki değerlerin korunup geri kalan bölgelere sıfır atılması ile gerçekleşir. Matematiksel olarak bir girdi maxrix P için, kodlayıcı aşağıdaki işlemi gerçekleştirir:



Şekil 4.5: CNN ve CAE modelleri için önerilmiş filtre birleştirme tekniği.

$$e_i = \sigma(P * F^n + b) \quad (4.22)$$

burada σ aktivasyon fonksiyonunu, $*$ 2 boyutlu konvolüsyon işlemi, F^n ise n . konvolüsyonel filtreyi ve b kodlayıcı sabit terimini temsil etmektedir. Uzaysal çözünürlüğü ağın başından sonuna sabit tutmak adına girdi matrisi P 'nin kenarlarına sıfırlar eklenir. Sonrasında kod çözücünde girdi sinyalini geri oluşturma fazı aşağıdaki şekilde gerçekleştirilir

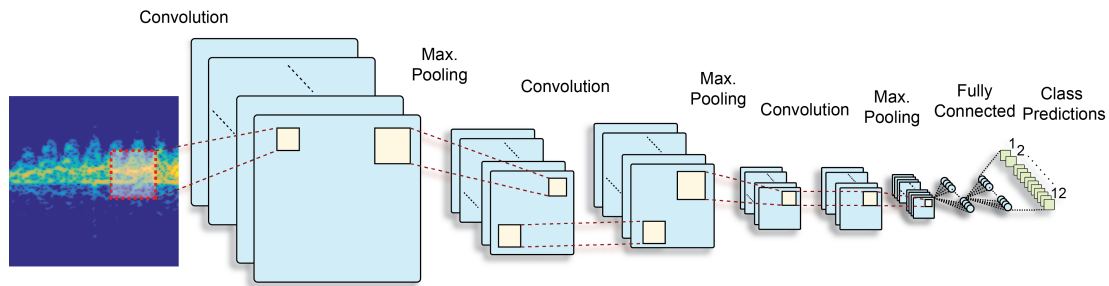
$$z_i = \sigma(e_i * \tilde{F}^n + \tilde{b}). \quad (4.23)$$

Burada z_i , i . girdinin geri oluşturulmuş halidir. \tilde{F}^n ise kod çözücü katmanın n . 2D dekonvolüsyonel filtresidir ve \tilde{b} ise kod çözücünün sabit terimidir. Burada gözetimsiz ön-eğitim işlemi AE'ye benzer bir şekilde aşağıdaki ortalama kare hatasını minimize ederek gerçekleştirilir:

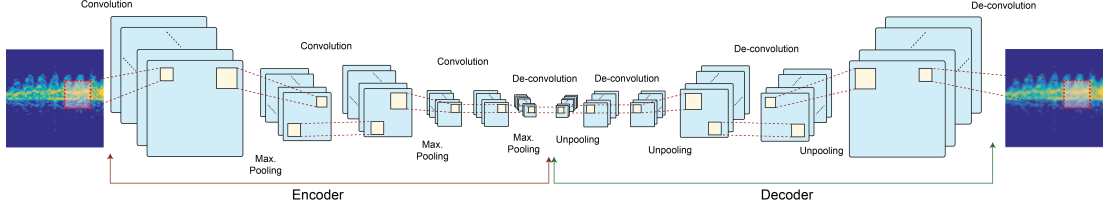
$$E(\theta) = \sum_{i=1}^m (x_i - z_i)^2 \quad (4.24)$$

AE'dekine benzer olarak gözetimsiz ön-eğitimden sonra, tersine-havuzlama ve dekonvolüsyonel katmanları içeren kod çözücü ağdan çıkartılır ve kodlayıcının sonuna tam bağlı katmanlar ve softmax sınıflandırıcısı eklenir. Sonrasında ağa, Denklem (4.19) optimize edilerek hassas ayar yapılır. Tez kapsamında CAE yapısında da tüm ağlardaki gibi Relu aktivasyon fonksiyonu kullanılmış olup optimizasyon işlemi ADAM ile yapılmıştır. Kodlayıcının sonuna 150 nöronlu 2 adet tam bağlı katman eklenmiş ve tam bağlı katmanda 0.5 olasılıkla bırakma işlemi uygulanmıştır.

Hiperparametrelerin optimizasyonu ızgara taraması ile yapılmış olup sonuçlar Çizelge 5.3'te verilmiştir. Sonuçlara göre en iyi sınıflandırma performansını veren CAE mimarisi seçilmiştir. Mimari 3 katmanlı, her katmanda 30 filtre içeren ve 3x3 ve 9x9'luk filtreleri birleştiren, CNN mimarisine benzer bir mimaridir. Genel CAE mimarisi Şekil 4.7 üzerinden görülebilir.



Şekil 4.6: Tez kapsamında önerilen CNN mimarisi her katmanda 30 3x3'lük filtre içeren 3 katmandan meydana gelmektedir. Sonunda da 2 adet 150 nöronluk tam bağlı katman içermektedir.



Şekil 4.7: Tez kapsamında önerilen CAE mimarisi. Gözetimsiz ön-eğitimden sonra kod çözücü çıkartılır ve kodlayıcının sonuna 2 adet tam bağlı katman ile softmax sınıflandırıcısı eklenir.

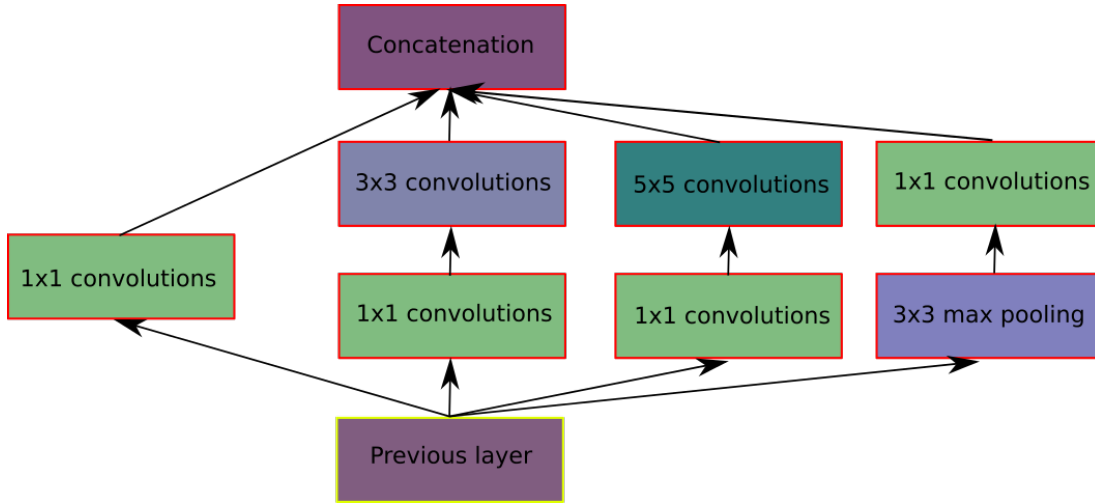
4.4.4 Aktarımlı Öğrenme

Aktarımlı öğrenme, öğrenilmiş bir modelin parametrelerinin, modelin eğitildiği veri ile alakalı bir başka problemin çözülmesi için kullanılmasıdır. Son yıllarda literatürde bu konuda birçok yayın mevcuttur. Bunlara örnekler olarak ses verisi üzerinden akustik sahne sınıflandırması [?] ve ultrason görsellerindeki akciğer fibrozununun tespiti için optik veriler ile başlatılan ağlar gösterilebilir [?]. Aktarımlı öğrenmenin altında yatan fikir tam olarak şu şekilde özetlenebilir: Ağırlıkların aktarıldığı model veri seti A'nın pek çok örneği kullanılarak eğitilmiş olduğundan, eğer veri seti B veri seti A'daki problemle ilişkili ise, veri seti B'nin az sayıda örneği ile model 'ince ayar' yapılarak eğitilebilir. Bunlara bağlı olarak aktarımlı öğrenme tekniği az sayıda etiketli veri olduğu durumlarda kullanılacak güçlü bir teknik olarak ortaya çıkmaktadır. Literatürde mikro-Doppler hareket sınıflandırma problemi üzerine de aktarımlı öğrenme uygulanmıştır.

Park [?], önceden ImageNet veri setiyle eğitilmiş AlexNet ve VGGNet modellerine ince ayar uygulayarak 5 değişik yüzme hareketini sadece 625 veri ile toplam mikro-Doppler veri setinden %80.3'lük (sıfırdan eğitim ile %66.7 performans elde edildiği raporlanmaktadır.) bir sınıflandırma doğruluğu ile sınıflandırmıştır. Aktarımlı öğrenmenin, çok az sayıda (<500) eğitim verisi olduğu durumlarda gözetimsiz ön-eğitim tekniğinden daha etkili olduğu gösterilmiştir [?].

Bu tez kapsamında, ImageNet üzerine eğitilmiş CNN modelleri olan VGGNet, GoogleNet [?] ve ResNet [?] modellerine, oluşturulan veri seti ile ince ayar uygulanarak mikro-Doppler verileri sınıflandırılmıştır. VGGNet 16 katmanlı bir CNN mimarisidir ve 2014 yılında ImageNet verisinin sınıflandırılması üzerine olan Büyük Ölçekli Görsel Tanıma Yarışması (ILSVRC)'de 2. en yüksek başarıyı almıştır. VGGNet ReLu aktivasyonları ve 3x3'lük konvolüsyonel filtreler kullanmaktadır. GoogleNet ILSVRC 2014'ün birincisi olan mimaridir. GoogleNet mimarisi yapı olarak alışlagelen CNN mimarilerinden biraz farklıdır. GoogleNet her biri kendi içinde birkaç modül içeren ve ismini Christopher Nolan'ın Inception(Başlangıç) filminden alan 9 adet Inception (ağ içinde ağ mantığı üzerine) modülünden oluşmaktadır. Her inception modülü tez kapsamında kullanılan CAE mimarisindeki benzer bir filtre birleştirme yapısı içermektedir. Fakat CAE'den farklı olarak her bir konvolüsyonel filtre öncesinde 1x1'lik konvolüsyonel filtreler ile verilen girdiyi işleyip renk bandında boyut indirgeme işlemi yapar. Inception bloğunun yapısı genel olarak Şekil 4.8 üzerinden görülebilir.

Tez kapsamında aktarımlı öğrenme uygulanan son popüler mimari Microsoft'un 2015



Şekil 4.8: GoogleNet mimarisinin yapı taşı olan inception bloğunun gösterimi [?].

yılında önerdiği ve ILSVRC 2015'in birincisi olan ResNet mimarisidir. ResNet mimarisi 2015 yılına kadar önerilmiş en derin mimaridir ve 152 katmandan oluşmaktadır. Derinlik arttıkça, beraberinde gelen 'bozunma' problemi, ResNet kapsamında önerilen Artçıl Öğrenme [?] yapısı ile çözülmüştür. Bozunma problemi doğrudan aşırı öğrenme ile alakalı değildir, ağın derinliğinin artmasıyla performansın sature olması ve ani bir şekilde düşmesi durumunun karşılığıdır [?]. Derin Artçıl Ağlar Şekil 4.9(a)'da gösterilen yapıtaşlarından meydana gelmektedir. Bu yapı doğrudan standart haritalama işlemi olan $y_l(x) := h(x_l)$ 'i hesaplamaktansa, şu artçıl haritalama işlemi hesaplar:

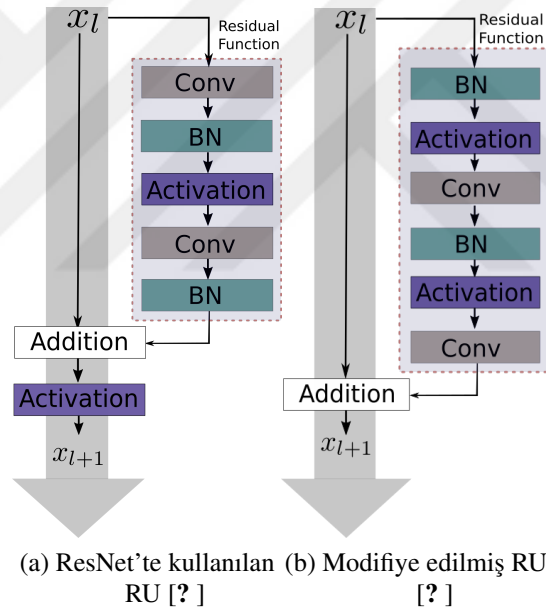
$$y_l = h(x_l) + F(x_l, W_l), \quad (4.25)$$

$$x_{l+1} = f(y_l), \quad (4.26)$$

burada x_l , l . artçıl ünite (RU)'nun girdisi, $W_l = W_{l,k} | 1 \leq k \leq K$. l . RU'nun ağırlık ve sabit terim değerleri, K RU'daki katman sayısı, F artçıl fonksiyonu (iki adet konvolüsyonel katmanın yığınlanması), $h(x_l) = x_l$ kestirme bağlantı kullanılarak sağlanan birim haritalama değerini ve son olarak f aktivasyon fonksiyonunu ifade etmektedir. Bunlara ek olarak ResNet Yığın Normalizasyonu (BN) katmanları da içermektedir. BN katmanlarında ağa her iterasyonda verilen yığınlar içsel eş-varyans kayması durumunu azaltmak için standardize (sıfır ortalama birim varyans) edilir [?]. BN bu sayede ağ eğitilirken yüksek öğrenme oranları kullanılabilmesine olanak tanımaktadır. Bahsedilen kestirme bağlantılar ağ içerisinde basitçe birim haritalama işlemleri vasıtasıyla uygulanabilir. Sıralı katmanların bu artçıl bağlantılar ile bağlanması ile, ağ her seferinde önceki katmanlarda öğrenilmemiş yeni bir öznelik yapısını öğrenmeye çalışır. Eğer yeni katman ağın başarımını düşürecek etkide bulunursa, veya ağın performansına etkide bulunmuyorsa, gradyan değeri kestirme bağlantı üzerinden aktarılır. Böylece artçıl öğrenme, çok derin ağların, ağın "çok derin" olup olmamasından endişelenmeksizin inşa edilebilmesine olanak tanımaktadır, çünkü ileri katmanlar fayda sağlamazsa RU'lar birim matrisi öğrenebilir ve bu ileri katmanlar ağın öğrenimine katkıda bulunmaz, dolayısıyla performansa etki etmez.

Bu tez kapsamında aktarımlı öğrenme şu şekilde uygulanmıştır: İlk olarak ImageNet üzerine eğitilen ağların tam bağlı katmanları ve softmax sınıflandırıcıları ağdan çıkar-

tılmıştır. Daha sonra, iki adet 150 nöron barındıran tam bağlı katman ile 12 nöronlu bir softmax sınıflandırıcısı ağa bağlanmıştır. Tam bağlı katmanlarda %50 ihtimalle dropout uygulanmıştır [?]. ImageNet Kırmızı-Yeşil-Mavi(RGB) görsellerden oluşturulan eldeki gri tonlama spektrogramlarının hepsi 3 kanala kopyalanıp RGB spektrogramlar oluşturulmuştur. Son olarak ince ayar, tüm ağların çok düşük bir öğrenme oranı kullanılarak eğitilmesiyle uygulanmıştır. İnce ayar esnasında 0.0001 öğrenme oranlı 100 yığın kullanan Stokastik Gradyan İnişi (SGD) algoritması kullanılmıştır. Burada ağın sonuna rastgele sayılarla başlatılmış ağırlıklar içeren tam bağlı katmanlar eklendiğinden, ince ayar esnasında yüksek öğrenme oranı kullanılmaması önemlidir. Nitekim eğitimin ilk iterasyonlarında rastgele başlatılan katmanlardan kaynaklı ortaya çıkacak hata, geri yayılım esnasında ilk katmanlarda öğrenilmiş filtreleri bozabilmektedir. Ayrıca ADAM tarzı adaptif öğrenme oranı uygulayan yöntemler de ilk katmanlarda öğrenilmiş filtreleri bozabilmektedir. Bu yüzden, hata çok düşük bir öğrenme oranı kullanan SGD algoritması kullanılarak iletilmelidir (Burada BN kullanımının, yüksek öğrenme oranları ile eğitime olanak sağlaması ile bir çelişki bulunmamaktadır. BN modellerin ilk eğitimleri aşamasında yüksek öğrenme oranları kullanımına olanak tanır; ince ayar esnasında mutlaka ufak öğrenme oranları kullanılmalıdır.)

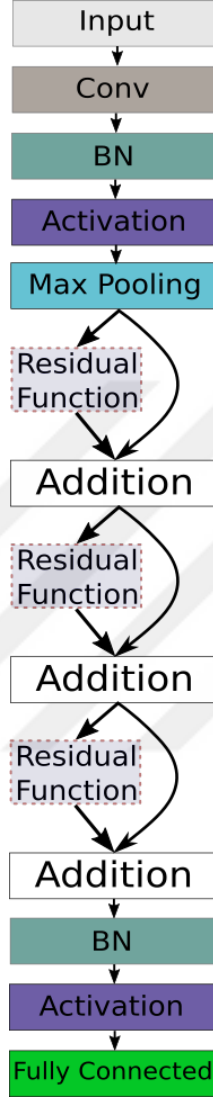


Şekil 4.9: Artçıl Ünitelerin yapısı (BN yığın normalizasyonunun kısaltmasıdır).

4.5 Mikro-Doppler Simülasyon Verisi Üzerinden Aktarımlı Öğrenme

Bu bölümde transfer alanı birçok mikro-Doppler hareketini ve varyantını içeren bir veri seti ile eğitilmiş bir model üzerinden aktarımlı öğrenme yapıldığında performans artışı olup olmadığı sorusu araştırılmıştır. Bu kapsamda Bölüm 2.2'da oluşturulmuş 32000 veriden oluşan 7 sınıflı veri seti ile eğitilmiş bir modelden aktarımlı öğrenme uygulanmıştır. Aktarımlı öğrenme için önerilen model, Artçıl Öğrenme üniteleri kullanan 30 katmanlı bir modeldir. Bu modelde, standart RU yapısının ResNet modeline göre biraz değiştirilmiş hali kullanılmıştır ve bu yapı Şekil 4.9(b) üzerinden görülebilir. Bu modelin standart RU yapısına göre daha kolay eğitilebilir olduğu ve daha genelleyci

olduğu literatürde belirtilmiştir [?]. Bu yapıda RU yine 3 konvolüsyonel katman içermektedir, fakat aktivasyon fonksiyonu, RU'nun sonundaki toplama işleminden önceye alınmıştır. Önerilen model (Bundan sonra DivNet olarak belirtilecektir) 3 adet modifiyeli RU'nun oluşturduğu toplamda 30 katmandan oluşmaktadır ve Şekil 4.10 üzerinde gösterilmiştir.



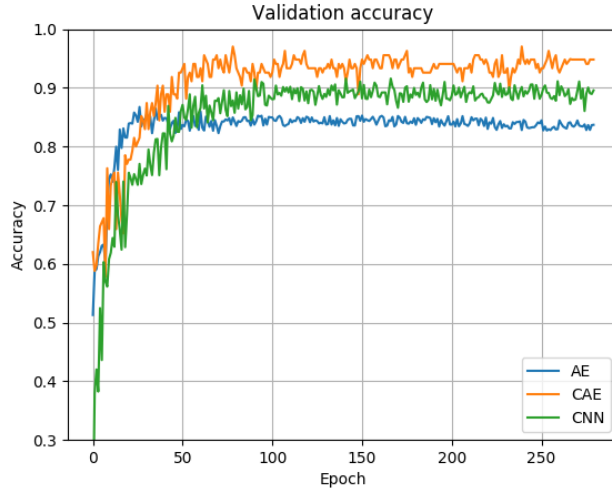
Şekil 4.10: Önerilen DivNet mimarisi.

5. DENEYSEL SONUÇLAR

Tez kapsamında, derin öğrenme mimarilerinin implementasyonu Python üzerinde arkaplanda Tensorflow [?] kütüphanesini kullanan Keras [?] kütüphanesi kullanılarak yapılmıştır. Bu bölümde belirtilen sınıflandırma başarımları karışıklık matrisinin diyagonal elemanlarının ortalaması şeklinde verilmiştir. Sınıflandırma başarımlarını kıyaslamak adına öncelikle geleneksel yaklaşım ile AE, CNN ve CAE modelleri kıyaslanacaktır. Sonrasında, ağırlık başlatma metodları kıyaslanacaktır. Bu kıyas, ImageNet üzerine eğitilmiş VGGNet, GoogleNet ve ResNet-50 modellerinden aktarımlı öğrenme ile gözetimsiz ön-eğitim arasında yapılacaktır. Daha sonra aktarımlı öğrenme metodları ile, önerilen DivNet yapısı detaylı bir şekilde kıyaslanacaktır.

5.1 Geleneksel Yaklaşım Karşı Derin Öğrenme

Bu bölümde, öncelikle, tüm derin öğrenme modellerinin hiper-parametreleri optimize edilmiştir. Çizelge 5.1-5.2-5.3 üzerinde çeşitli parametreler için DNN yapılarının elde ettiği ortalama başarımlar oranları gösterilmiştir. AE yapısı için 2 parametre hesaba katılmıştır: Katmanlarda kullanılan nöron sayısı ve derinlik. CNN ve CAE yapıları içinse 3 parametre hesaba katılmıştır: Derinlik, filtre boyutu ve katman başına nöron sayısı. Bu değerler incelendiğinde, tüm modeller için 3 katmandan sonra aşırı öğrenme gözlemlenmiş, 3 katmana kadar ise öğrenememe durumu gözlemlenmiştir.



Şekil 5.1: AE, CNN ve CAE için validasyon performans kıyaslaması.

Tüm DNN'ler için ortalama sınıflandırma doğruluğu hesaplanırken veriler %80 eğitim ve %20 test verisi olarak ayrıştırılmıştır. Sonrasında eğitim verisinin içinden rastgele %20'lik kısım her iterasyonda validasyon seti olarak belirlenip, modelin iterasyonlara bağlı değişim grafiği elde edilmiştir. Bu işlem toplam 10 kere tekrar edilip ortalama alınarak bu bölümdeki sonuçlar elde edilmiştir. Şekil 5.2 üzerinde modellerin validasyon performansı iterasyon bazında gösterilmiştir. Burada gözetimsiz ön-eğitimin etkisi ortaya çıkmaktadır: Yalnızca birkaç iterasyon içinde AE ve CAE modelleri %51 and %62, sınıflandırma doğruluğuna ulaşırken, CNN yalnızca %31 sınıflandırma doğruluğu sağlamaktadır. Gözetimsiz ön-eğitim esnasında AE ve CAE ağırları toplamda 3 iterasyon boyunca eğitilmiş ve ortalama kare hatası olarak AE 0.5, CAE ise 0.01 değerlerini vermektedir. Validasyon performansının tüm yöntemler için 75 iterasyondan sonra aşağı yukarı sabit kaldığı, tüm yöntemler için aşırı öğrenme yapılmadığını doğrulamaktadır.

Sınıflandırma performanslarını özetleyen karışıklık matrisleri geriye bırakılan test verisi üzerinden eğitilen modellerin test edilmesiyle, SVM, AE, CNN, CAE için sırasıyla Çizelge 5.4-5.7 arasında verilmiştir. Bu tablolarda yeşil doğru sınıflandırılan veri oranını, kırmızı >%20'den fazla sınıflandırma hatasını, kahverengi %10 ve %20 arasındaki sınıflandırma hatasını ve son olarak sarı %10'dan düşük sınıflandırma hatasını belirtmektedir. Bu sonuçlar doğrultusunda, DNN'lerin geleneksel yöntemlere göre fazladan %7 ila %17 sınıflandırma doğruluğu sağladığı görülmektedir. Sonuçlar özellikle birbirine benzeyen mikro-Doppler imzalarının ayrıştırılması için özneliklerin doğrudan veriden öğrenilmesinin, el ile çıkartılmasına göre fayda sağladığını doğrulamaktadır.

DNN mimarileri arasında CAE, sağladığı %94.2'lik sınıflandırma doğruluğu ile ön plana çıkmaktadır. CNN ve AE ise sırasıyla %90.1 ve %84.1 sınıflandırma doğruluğu sağlamıştır. Bu sonuçlar, CNN ve AE'nin lokal öznelikleri öğrenme ve gözetimsiz ön-eğitim yeteneklerini tek mimaride birleştirmiş CAE'nin daha üstün olduğunu göstermektedir.

5.1.1 Sınıflandırma sonuçlarının tartışılması

AE mimarisi birbirine benzer sınıflar arasında oldukça yüksek miktarda karışıklıklar sergilemiştir. Bu sınıflar arasında en çok:

- Baston ile yürüme Topallama ve Yürüteç ile yürümeyle,
- Düşme Sandalyeden düşme ve oturmayla,
- Emekleme Sürünmeyle

karışmaktadır. Burada sürünme ile koşmanın karışması şaşırtıcıdır. CNN, AE'ye göre sandalyeden düşme ve oturma hareketlerini ayırt etmekte daha başarılı olarak gözlemlenirken, düşme ve sandalyeden düşme hareketinin başarımı AE'ye göre oldukça yüksektir. CAE modelinin ise neredeyse tüm sınıfları yüksek başarımla sınıflandırdığı, yalnızca emekleme hareketinde düşük performans sergilediği gözlemlenmiştir. Sürünme

hareketinin emekleme hareketiyle bu kadar karışıklık yaratmasının sebebi olarak, askeri stil sürünme hareketini sergilemenin zor olması, dolayısıyla hareketin yapılışı esnasında karın yere değmediği için emekleme sınıfına benzemesi gösterilebilir. Bu da doğrudan, sınıf içi varyansın yüksek olduğu sürünme hareketini genellemenin zorluğunu göstermektedir.

5.1.2 Darboğaz özneliklerinin performansı

Gözetimsiz ön-eğitimin önemini vurgulamak adına CAE üzerinden bir deney yapılmıştır. Bu deneyde 2 model ele alınmıştır:

1. Ağırlıkları rastgele başlatılan bir CNN
2. Gözetimsiz ön-eğitim ile 20 iterasyon eğitilmiş CAE

CNN için, ağın sonundaki tam bağlı katman ve softmax sınıflandırıcısı çıkartılmış ve ağ hiç eğitilmeksizin rastgele ağırlıklar ile darboğaz öznelikleri elde edilmiştir. İkincil olarak CAE yapısı 20 iterasyon gözetimsiz ön-eğitim ile eğitilmiş, sonrasında kod çözücü kısmı ağdan çıkartılmış ve ağ darboğaz öznelikleri elde etmek için kullanılmıştır. İki öznelik seti de bir SVM sınıflandırıcısı ile sınıflandırılmıştır: Deneyin sonucunda rastgele başlatılan CNN modelinin %8.3 (12 sınıf olduğu için rastgele seçmeye tekabül ediyor) başarımla sağladığı görülürken, CAE'nin %83.4 (El işçiliği ile elde edilen performanstan daha yüksek) başarımla sağladığı görülmüştür. Dolayısıyla, gözetimsiz ön-eğitim algoritmasının ağ ağırlıkları için iyi bir başlangıç sağladığı kesin olarak söylenebilir.

Çizelge 5.1: AE için parametre optimizasyon çizelgesi.

| Derinlik | Genişlik | Perf. (%) | Derinlik | Genişlik | Perf. (%) |
|----------|-------------------|-------------|----------|---------------------------|-----------|
| 1 | 20 | 74.1 | 4 | 20-50-100-200 | 83.4 |
| 1 | 50 | 76 | 4 | 40-100-200-400 | 83.1 |
| 1 | 100 | 75.5 | 4 | 100-200-400-800 | 79.8 |
| 2 | 20-50 | 79.8 | 5 | 20-50-100-200-400 | 82.7 |
| 2 | 50-100 | 81.1 | 5 | 40-100-200-400-800 | 82.8 |
| 2 | 100-200 | 80.9 | 5 | 100-200-400-800-1600 | 79.5 |
| 3 | 20-50-100 | 83.4 | 6 | 20-50-100-200-400-800 | 80.1 |
| 3 | 40-80-160 | 83.2 | 6 | 40-100-200-400-800-1600 | 78.9 |
| 3 | 50-100-200 | 84.1 | 6 | 100-200-400-800-1600-3200 | 78.3 |

5.2 Ağırlık Başlatma Yöntemlerinin Kıyaslanması

Bu bölümde DNN başlatma teknikleri kıyaslanmıştır. AE performans olarak CAE'nin gerisinde kaldığı için aktarımlı öğrenme modelleriyle yalnızca CAE kıyaslanmıştır. Bu bölümde de Bölüm 5.1'de uygulanan eğitim/doğrulama/test metodolojisi uygulanmıştır.

Burada, tüm ağ başlatma metodlarının ölçüt noktası (baseline) olarak kullanılan ağırlıkları rastgele başlatılmış, 3 katmanlı, filtre birleştirme uygulamayan bir CNN'den

Çizelge 5.2: CNN için parametre optimizasyon çizelgesi.

| Filtre Boyu | Derinlik | Genişlik | Perf(%) | Derinlik | Genişlik | Perf(%) | Derinlik | Genişlik | Perf(%) |
|-------------|----------|----------|---------|----------|-----------|-------------|----------|----------|---------|
| 3x3 - 9x9 | 1 | 5 | 80.2 | 3 | 5 | 88.4 | 5 | 5 | 87.2 |
| | 1 | 30 | 81.4 | 3 | 30 | 90.1 | 5 | 30 | 88.1 |
| | 1 | 100 | 81.5 | 3 | 100 | 90 | 5 | 100 | 88.4 |
| | 2 | 5 | 82.1 | 4 | 5 | 87.6 | 6 | 5 | 84.9 |
| | 2 | 30 | 86.2 | 4 | 30 | 89.9 | 6 | 30 | 85.5 |
| | 2 | 100 | 87.1 | 4 | 100 | 89.5 | 6 | 100 | 86.8 |
| 2x2 - 7x7 | 1 | 5 | 79.6 | 3 | 5 | 84.9 | 5 | 5 | 87.8 |
| | 1 | 30 | 80.4 | 3 | 30 | 88.8 | 5 | 30 | 88 |
| | 1 | 100 | 81.7 | 3 | 100 | 89.5 | 5 | 100 | 88.6 |
| | 2 | 5 | 83.4 | 4 | 5 | 86.9 | 6 | 5 | 85.3 |
| | 2 | 30 | 86.1 | 4 | 30 | 88.8 | 6 | 30 | 86.1 |
| | 2 | 100 | 87.6 | 4 | 100 | 89.3 | 6 | 100 | 86.2 |

Çizelge 5.3: CAE için parametre optimizasyon çizelgesi.

| Filtre Boyu | Derinlik | Genişlik | Perf(%) | Derinlik | Genişlik | Perf(%) | Derinlik | Genişlik | Perf(%) |
|-------------|----------|----------|---------|----------|-----------|-------------|----------|----------|---------|
| 3x3 - 9x9 | 1 | 5 | 83.8 | 3 | 5 | 92.2 | 5 | 5 | 93.2 |
| | 1 | 30 | 84.9 | 3 | 30 | 94.2 | 5 | 30 | 93.4 |
| | 1 | 100 | 84.7 | 3 | 100 | 93.2 | 5 | 100 | 93.9 |
| | 2 | 5 | 88.2 | 4 | 5 | 94 | 6 | 5 | 90.1 |
| | 2 | 30 | 88.4 | 4 | 30 | 94.1 | 6 | 30 | 90.9 |
| | 2 | 100 | 89 | 4 | 100 | 93.7 | 6 | 100 | 90.3 |
| 2x2 - 7x7 | 1 | 5 | 82.1 | 3 | 5 | 92.4 | 5 | 5 | 92.1 |
| | 1 | 30 | 84.4 | 3 | 30 | 93.9 | 5 | 30 | 92.9 |
| | 1 | 100 | 84.5 | 3 | 100 | 93.8 | 5 | 100 | 91.2 |
| | 2 | 5 | 88 | 4 | 5 | 92.8 | 6 | 5 | 89.8 |
| | 2 | 30 | 88.5 | 4 | 30 | 93.2 | 6 | 30 | 90.5 |
| | 2 | 100 | 88.6 | 4 | 100 | 92.9 | 6 | 100 | 90.3 |

yüksek başarımlı olduğu görülmüştür. Ayrıca CAE modelinin, Imagenet üzerine eğitilmiş VGGNet ve GoogleNet modellerine göre daha iyi performans verdiği görülmüştür. Modellerin genel validasyon performansları Şekil 5.2 üzerinden görülebilir.

Ayrıca ağ başlatma modellerinin eğitim veri sayısına bağlı performans değeri de incelenmiştir. Bunun için tüm verinin %20'lik kısmı test verisi olarak ayrılmış (tüm sınıflardan eşit sayıda veri içeren) ve geri kalan eğitim verisi 160 veriden başlayarak 806 veriye kadar artırılarak modellerin performansları incelenmiştir. Bu teknik 10 sefer tekrar edilip sonuçların ortalaması Şekil 5.3 üzerinden görülebilir. Burada aktarımlı öğrenmenin, çok az sayıda eğitim verisi olduğu durumlarda (<200 veri) ağırlıkları rastgele başlatan modele göre %20 daha yüksek başarımlı olduğu görülmektedir. Fakat 600 örnekten sonra aktarımlı öğrenmenin sağladığı performans sabitlenmiştir. Buna karşın, CAE modelinin performansını eğitim verisi sayısına göre istikrarlı bir şekilde arttırdığı görülmüştür.

Bu nedenle, eğitim veri sayısı çok az olduğu durumlarda aktarımlı öğrenme bariz avantajlara sahipken, veri sayısı bir miktar arttırıldığında CAE modeli çok daha az hesaplama karmaşıklığına ihtiyaç duyar hem de daha yüksek performans gösterir. Modellerin hesaplama karmaşıklıkları Bölüm 5.4'te anlatılan Çizelge 5.12 üzerinde verilmiştir. Nitekim CAE öznelikleri doğrudan mikro-Doppler verileri üzerinden öğrenirken, VGGNet ve GoogleNet, ilgisiz görsellerden öğrenilmiş genel şekilsel öznelikleri az miktarda değiştirerek kullanmaktadır. Bu da aktarımlı öğrenmeyi, gürültü veya diğer arzulanmayan etkiler içeren sinyallere daha az gürbüz hale getirmektedir. Bu etki bir sonraki bölümde incelenecektir.

Çizelge 5.4: SVM için elde edilen karışıklık matrisi (ortalama doğruluk %76.9).

| % | Yürüme | T. Sandalye | Topallama | Baston | Yürüteç | Düşme | K. Değneği | Emekleme | Sürünme | Koşma | Oturma | S. Düşme |
|-------------|--------|-------------|-----------|--------|---------|-------|------------|----------|---------|-------|--------|----------|
| Yürüme | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T. Sandalye | 0 | 83.9 | 3.2 | 12.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Topallama | 0 | 0 | 72.4 | 15.6 | 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Baston | 0 | 3.8 | 0 | 76.9 | 19.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Yürüteç | 0 | 4.4 | 0 | 8.7 | 86.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Düşme | 25 | 0 | 0 | 0 | 0 | 55 | 0 | 0 | 0 | 0 | 0 | 20 |
| K. Değneği | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| Emekleme | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 37.5 | 12.5 | 0 | 0 | 0 |
| Sürünme | 0 | 14.3 | 0 | 0 | 0 | 0 | 0 | 43.1 | 28.3 | 14.3 | 0 | 0 |
| Koşma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 95.7 | 0 | 4.3 |
| Oturma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 |
| S. Düşme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14.3 | 85.7 |

Çizelge 5.5: AE için elde edilen karışıklık matrisi (ortalama doğruluk %84.1).

| % | Yürüme | T. Sandalye | Topallama | Baston | Yürüteç | Düşme | K. Değneği | Emekleme | Sürünme | Koşma | Oturma | S. Düşme |
|-------------|--------|-------------|-----------|--------|---------|-------|------------|----------|---------|-------|--------|----------|
| Yürüme | 91.1 | 0 | 0 | 0 | 0 | 0 | 0 | 8.9 | 0 | 0 | 0 | 0 |
| T. Sandalye | 0 | 91.4 | 2.9 | 0 | 4.1 | 0 | 0 | 1.6 | 0 | 0 | 0 | 0 |
| Topallama | 0 | 0 | 84.4 | 7.3 | 8.3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Baston | 0 | 0 | 10.1 | 69.1 | 20.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Yürüteç | 0 | 0.1 | 0.1 | 9.1 | 90.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Düşme | 0.5 | 0 | 0 | 0 | 0 | 89.3 | 0 | 0 | 0 | 0 | 0 | 10.2 |
| K. Değneği | 7 | 0 | 0 | 0 | 0 | 0 | 91.5 | 0 | 1.5 | 0 | 0 | 0 |
| Emekleme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 70.4 | 29.6 | 0 | 0 | 0 |
| Sürünme | 2.1 | 0 | 0 | 0 | 0 | 0 | 0 | 24.5 | 60.9 | 12.5 | 0 | 0 |
| Koşma | 0.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 98.1 | 0 | 1.5 |
| Oturma | 0 | 0 | 0 | 0 | 0 | 19.6 | 0 | 0 | 0 | 0 | 80.4 | 0 |
| S. Düşme | 0 | 0 | 0 | 0 | 0 | 8.5 | 0 | 0 | 0 | 0 | 0 | 91.5 |

5.2.1 Aktivasyon Haritalarının incelenmesi

DNN'ler tarafından öğrenilen özniteliklerin anlaşılması için, verilen spektrogram görselleri için çeşitli katmanlardan elde edilen aktivasyon haritaları incelenmiştir. Karşılaştırma iki farklı düşme spektrogram görüntüsü üzerinden yapılmıştır: Nispeten temiz ve gürültüsüz bir spektrogram ve 'kirli' bir spektrogram. Şekil 5.4'te nispeten 'temiz' bir spektrogram görselinin tüm DNN'lerin 1., 3., 5. ve son katmanı sonrası elde edilen aktivasyon haritası paylaşılmıştır. CAE modelinin son katmanından elde edilmiş görselin (darboğaz öznitelikler) gerçek sınıf gösterimine en yakın olduğu gözlemlenmiştir. Buna ek olarak CAE'nin sınıfı ayırt etmekte kullanılacak olan üst zarfa odaklandığını, VGGNet'in ise hareketi modellemek yerine alt zarfa odaklandığı ve arkaplan yansımasını modellemeye çalıştığı gözlemlenmektedir.

İkinci senaryoda DNN'lerin daha gürültülü spektrogram görselleri altında aktivasyon haritaları incelenmiştir. Örnek olarak seçilen spektrogram görseli, USRP üzerinden veri alınırken yapılan bir hatadan kaynaklanan dikey bir 'sivri uç' ve görüntü içermektedir. Bu veri Bölüm 2'de bahsedilen 1007 veriden oluşan veri setine dahil edilmemiş, ancak ilginç bir örnek oluşturduğu için eğitilmiş model üzerinden aktivasyon haritası hesaplanmıştır. Bu senaryoda elde edilen aktivasyon haritaları Şekil 5.5 üzerinde gösterilmiştir. Burada özellikle 5. katmanda GoogleNet modelinin yüksek oranda gürültüyü öğrendiğini, CNN ve VGGNet'in ise bahsedilen 'dikey sivri uç' öğrendiğini gözlemlemek mümkündür. Buna karşın CAE yönteminin gürültü ve istenmeyen sinyaller içeren spektrogramlar için bile daha gürbüz öznitelikler öğrendiği açıktır.

Çizelge 5.6: CNN için elde edilen karışıklık matrisi (ortalama doğruluk %90.1).

| % | Yürüme | T. Sandalye | Topallama | Baston | Yürüteç | Düşme | K. Değneği | Emekleme | Sürünme | Koşma | Oturma | S. Düşme |
|-------------|--------|-------------|-----------|--------|---------|-------|------------|----------|---------|-------|--------|----------|
| Yürüme | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T. Sandalye | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Topallama | 0 | 0 | 86.2 | 13.8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Baston | 0 | 0 | 3.9 | 81.9 | 14.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Yürüteç | 0 | 0 | 0 | 10.3 | 89.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Düşme | 0 | 0 | 0 | 0 | 0 | 86.4 | 0 | 0 | 0 | 0 | 0 | 13.6 |
| K. Değneği | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| Emekleme | 0 | 0 | 0 | 0 | 0 | 0 | 10.3 | 51.2 | 30.8 | 7.7 | 0 | 0 |
| Sürünme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 85.7 | 14.3 | 0 | 0 |
| Koşma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| Oturma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 |
| S. Düşme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

Çizelge 5.7: CAE için elde edilen karışıklık matrisi (ortalama doğruluk %94.2).

| % | Yürüme | T. Sandalye | Topallama | Baston | Yürüteç | Düşme | K. Değneği | Emekleme | Sürünme | Koşma | Oturma | S. Düşme |
|-------------|--------|-------------|-----------|--------|---------|-------|------------|----------|---------|-------|--------|----------|
| Yürüme | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T. Sandalye | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Topallama | 0 | 0 | 93.1 | 6.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Baston | 0 | 0 | 0 | 91.1 | 8.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Yürüteç | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Düşme | 0 | 0 | 0 | 0 | 0 | 89 | 0 | 0 | 0 | 0 | 9.4 | 1.6 |
| K. Değneği | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| Emekleme | 4.9 | 0 | 0 | 0 | 0 | 0 | 7.2 | 65.5 | 22.4 | 0 | 0 | 0 |
| Sürünme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8.5 | 91.5 | 0 | 0 | 0 |
| Koşma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| Oturma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 |
| S. Düşme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |

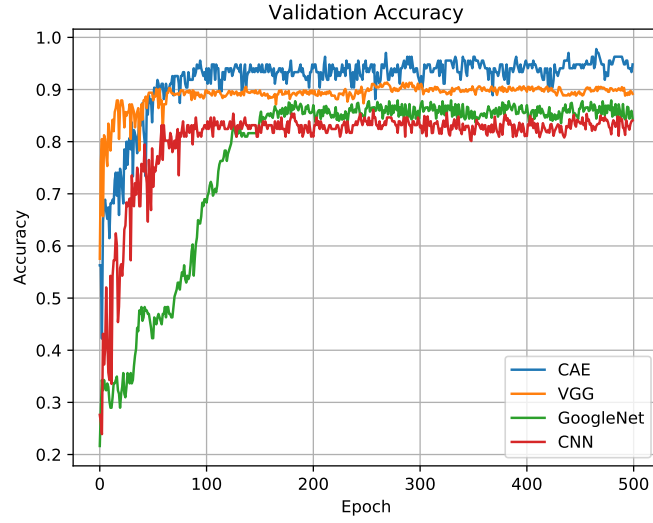
5.3 Önerilen Yöntemle Aktarımlı Öğrenmenin Kıyaslanması

Bu bölümde, Bölüm 2.2.1’de bahsedilmiş simülasyon verileri üzerine eğitilmiş artçıl aktarımlı öğrenme metodu Divnet’in, ImageNet üzerine eğitilmiş VGGNet, Resnet-50 ile ağırlıkları rastgele başlatılmış 4 katmanlı standart bir CNN ile kıyaslaması çeşitli Sinyal-Gürültü oranı (SNR) değerleri için yapılmıştır. Bu bölümün ilk aşamasında, aktarımlı öğrenme yapılan transfer alanının problem ile ilgili olmasının, ilgisiz bir veri seti ile eğitilmiş modellere göre avantajını göstermek adına ilk olarak darboğaz özneliklerinin performansı değerlendirilmiştir. Sonrasında çeşitli SNR değerleri için VGGNet, DivNet, ResNet-50 ve CNN modelleri kıyaslanmış ve sonuçlar paylaşılmıştır.

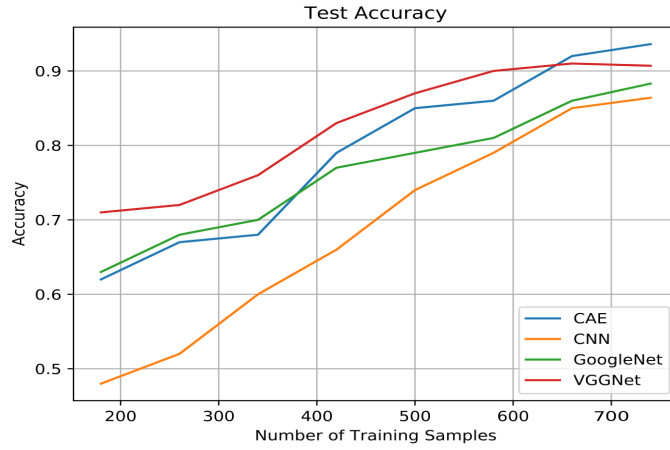
5.3.1 Darboğaz Özneliklerin kıyaslanması

Bölüm 5.1.2’dekine benzer bir şekilde ağ başlatımlarında, aktarım alanının (ImageNet ve Simülasyon Verileri) etkisini görmek adına VGGNet, ResNet ve Divnet yöntemlerinin öğrendiği darboğaz öznelikler kıyaslanmıştır. Burada darboğaz özneliklerinin performansını daha iyi görebilmek adına, ağların sonundaki tam bağlı katmanlar çıkartılmış, ağların tam bağlı katmana kadar olan tüm katmanları dondurulmuş(geri yayılımdan etkilenmemesi sağlanmış), ağların sonuna rastgele başlatılmış tam bağlı katmanlar eklenmiş ve yalnızca aktarım alanından öğrenilen özneliklerin doğrusal olmayan kombinasyonlarının tam bağlı katmanlarda öğrenilmesi 7 sınıflı (simülasyon verisindeki sınıf sayısı) ile gerçek radar verisi sınıflandırılmıştır. Tüm ağlar, 0.001 öğrenme oranlı stokastik gradyan inişi kullanılarak eğitilmiştir.

Bu deney sonucu elde edilmiş validasyon doğruluğu sonuçları Şekil 5.6 üzerinde gösterilmiştir. Optik görsellerden oluşan Imagenet üzerine eğitilmiş VGGNet ve ResNet’e göre, yalnızca mikro-Doppler simülasyon verileriyle eğitilmiş DivNet modelinin öğ-



Şekil 5.2: Ağırlık başlatma tekniklerinin performansı.

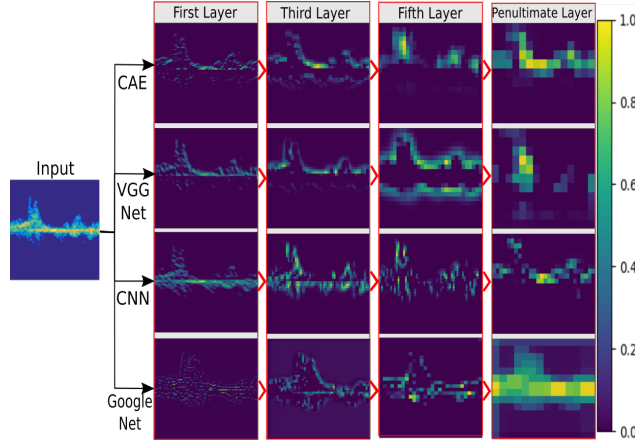


Şekil 5.3: Ağırlık başlatma tekniklerinin eğitimde kullanılan örnek sayısına göre performansı.

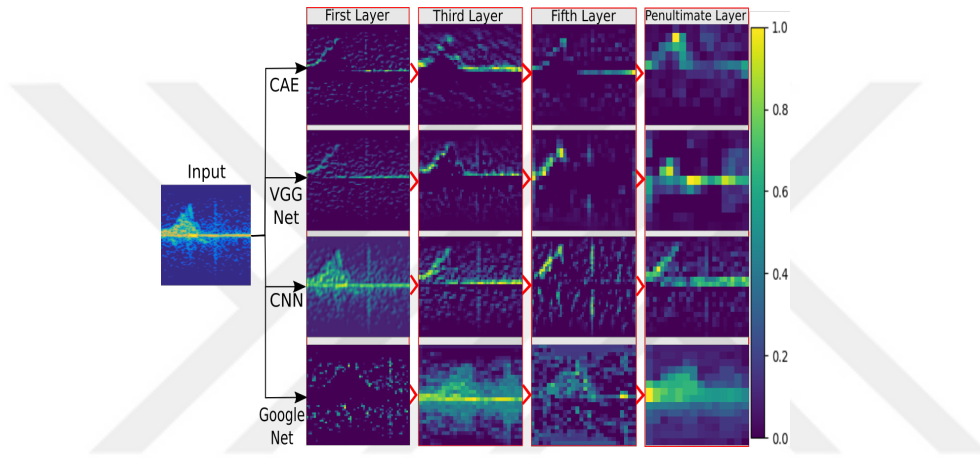
rendiği özneliklerin üstünlüğü açıkça görülmektedir.

5.3.2 Önerilen yöntemle aktarımlı öğrenmenin gürültü altında kıyaslanması

Bu bölümde tüm yöntemler için gürültü altındaki başarımlar incelenmiştir. Bunun için iki ayrı senaryo ele alınmıştır: 40-30-15 dB değerleri için 7 sınıf problemi, 40-30-15 dB değerleri için 12 sınıf problemi. 7 sınıf problemi için bütün SNR değerlerinde validasyon doğruluğu grafiği Şekil 5.7 üzerinde görülebilir. Burada DivNet'in yüksek SNR değerleri için (30 ve 40 dB) diğer yöntemlere göre daha yüksek performansa sahip olduğu ve 40. iterasyonda doyuma ulaştığı ve sonrasında da sabit kaldığı görülmektedir. Gürültü oranı görece yüksek olan 15 dB durumunda ise tüm aktarımlı öğrenme teknikleri aşağı yukarı aynı performansı vermektedir ancak hepsi sıfırdan eğitilmiş CNN'e göre %15 daha yüksek performans vermektedir. Böylelikle, aktarımlı öğrenme teknik-



Şekil 5.4: Düşme sınıf verisi için aktivasyon haritaları.



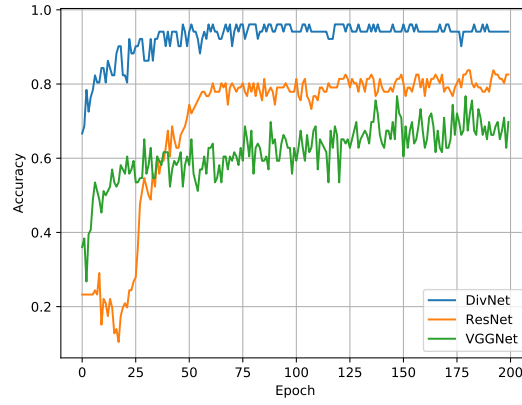
Şekil 5.5: İstenmeyen bileşenler barındıran düşme sınıf verisi için aktivasyon haritaları.

lerinin gürültüye karşı, ağırlıkların rastgele başlatılmasına göre daha gürbüz olduğu görülmektedir. Ancak, 50 katman kullanan ResNet'in 16 katman kullanan VGGNet modeline göre neredeyse aynı performansı gösteriyor olması, aktarımlı öğrenme yapılan alanın, çözülmekte olan problem ile alakasız olmasının açık bir sonucudur.

7 sınıf durumunda tüm ağlar için test başarımları da validasyon başarımlarına oldukça yakın olmaktadır ve bu da aslında başarılı bir öğrenme yapıldığının ölçütüdür. 7 sınıf durumu için Çizelge 5.10 üzerinden tüm SNR değerleri için ortalama test başarımları görülebilir. Önerilen yöntemin yüksek SNR değerlerinde diğer yöntemlere göre daha performanslı olduğu aşıkardır ve tüm aktarımlı öğrenme yöntemlerinin ağ ağırlıkları rastgele başlatılan CNN'e göre daha yüksek başarımlar elde ettiği açıkça görülmektedir.

En yüksek performansı veren DivNet modeli için 40 dB'deki 7 sınıflı karışıklık matrisi Çizelge 5.8 üzerinden görülebilir. En büyük sınıflandırma hatası bastonla yürüme ve yürüteçle yürüme sınıfları arasındadır. Bu durum şaşırtıcı değildir; nitekim bu iki sınıf, 7 sınıf probleminde birbirine en benzeyen sınıflardır. Ayrıca simülasyon veri seti yalnızca insan vücudundan kaynaklanan yansıma sinyallerini modellemektedir.

12 sınıf durumunda ise validasyon performansları Şekil 5.8 ve test performansları Çizelge 5.11 üzerinde gösterilmiştir. 15 dB'de VGGNet'in, çok daha derin olan Res-



Şekil 5.6: 40dB için darboğaz özneliklerinin kıyaslanması.

Çizelge 5.8: 40 dB verileriyle 7 sınıf probleminde DivNet için elde edilen karışıklık matrisi (ortalama doğruluk %97).

| % | Yürüme | Koşma | Topallama | Düşme | Oturma | Baston | Yürüteç |
|-----------|--------|-------|-----------|-------|--------|--------|---------|
| Yürüme | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| Koşma | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| Topallama | 0 | 0 | 97.2 | 0 | 0 | 2.8 | 0 |
| Düşme | 0 | 0 | 0 | 100 | 0 | 0 | 0 |
| Oturma | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| Baston | 0 | 0 | 0 | 0 | 0 | 96.4 | 3.6 |
| Yürüteç | 0 | 0 | 0 | 0 | 0 | 15.41 | 84.59 |

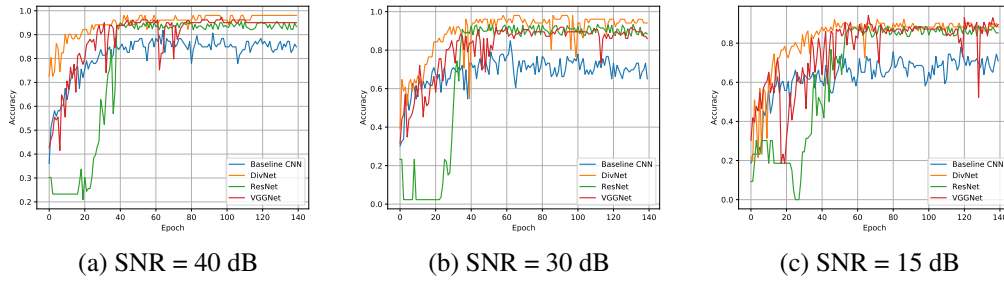
Net'ten daha iyi performans göstermesi, gürültü içeren görseller için daha 'az karmaşık' modellerin kullanılması gerektiğinin bir işaretidir. Son olarak 40 dB durumunda DivNet için karışıklık matrisi Çizelge 5.9 üzerinden incelendiğinde en benzer sınıflar olan yürüteçle yürüme/bastonla yürüme ve emekleme/sürünme sınıflarının karıştığı görülmektedir.

5.4 İşlem Karmaşıklığı

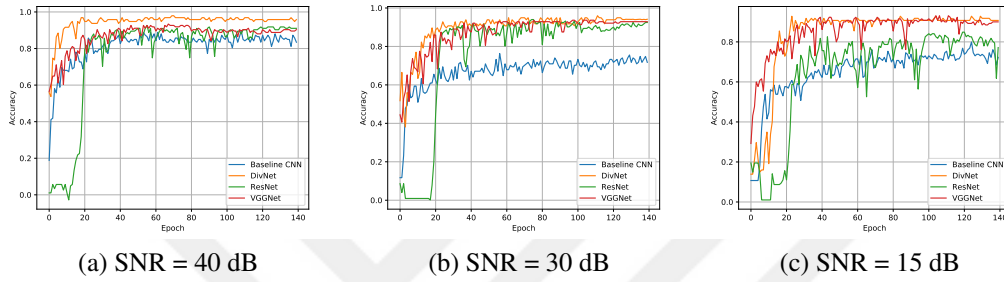
Tüm yöntemler için hesaplama süresi ve ağ parametreleri Çizelge 5.12 üzerinden görülebilir. Burada geleneksel yöntemi temsil eden SVM dışında diğer tüm yöntemler GPU üzerinde işlenmiştir. Modellerin eğitildiği sistemin spesifikasyonları şöyledir:

Çizelge 5.9: 40 dB verileriyle 12 sınıf probleminde DivNet için elde edilen karışıklık matrisi (ortalama doğruluk %96).

| % | Yürüme | T. Sandalye | Topallama | Baston | Yürüteç | Düşme | K. Değneği | Emekleme | Sürünme | Koşma | Oturma | S. Düşme |
|-------------|--------|-------------|-----------|--------|---------|-------|------------|----------|---------|-------|--------|----------|
| Yürüme | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| T. Sandalye | 0 | 95.6 | 4.4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Topallama | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Baston | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Yürüteç | 0 | 0 | 0 | 13.31 | 86.69 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Düşme | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| K. Değneği | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| Emekleme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 81.7 | 18.3 | 0 | 0 | 0 |
| Sürünme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11.4 | 88.6 | 0 | 0 | 0 |
| Koşma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 |
| Oturma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 |
| S. Düşme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |



Şekil 5.7: 7 Sınıf için validasyon performansı.



Şekil 5.8: 12 Sınıf için validasyon performansı.

Çizelge 5.10: Tüm yöntemler için 7 sınıfta test doğruluğu.

| Method | 40 dB | 30 dB | 15 dB |
|-----------------|-------|-------|-------|
| Proposed DivNet | 97% | 89% | 85% |
| ResNet | 95% | 88% | 85% |
| VGGnet | 94% | 87% | 86% |
| Baseline CNN | 86% | 70% | 69% |

Çizelge 5.11: Tüm yöntemler için 12 sınıfta test doğruluğu.

| Method | 40 dB | 30 dB | 15 dB |
|-----------------|-------|-------|-------|
| Proposed DivNet | 96% | 90.3% | 87% |
| ResNet | 91.1% | 90.5% | 82% |
| VGGnet | 91.2% | 89% | 87% |
| Baseline CNN | 86.7% | 75% | 70% |

NVIDIA Tesla K80 (24 GB VRAM), 378 GB RAM, Intel Xeon E5 2683 işlemci. SVM için raporlanan süreye özniteliklerin çıkartılması, öznitelik seçimi ve sınıflandırıcının eğitilmesi dahildir ve tüm işlemler CPU üzerinde yapılmıştır. Buna karşın, derin öğrenme mimarilerini eğitimi çok yüksek miktarda paralel işlem gücü gerektirmektedir ve bunun için GPU kullanımı oldukça önemlidir. Nitekim CPU üzerinde tüm DNN mimarilerinin eğitimi belirtilen sürelerden 10 kat daha uzun sürmektedir. Tüm derin öğrenme mimarilerinde toplam süre modellerin 280 iterasyon boyunca eğitilmesi ile verilmiştir. Konvolüsyonel mimarilerin kullandığı konvolüsyonel nöronlar, AE'nin kullandığı tekil nörona göre hesaplanma zorluğu olarak oldukça karmaşıktır. Ancak AE'deki nöronlar tam bağlı olduğu için toplamda eğitilmesi gereken daha fazla ağırlık

Çizelge 5.12: Tüm yöntemler için hesaplama karmaşıklığı değerleri

| | SVM | AE | CNN | CAE | GoogleNet | VGGNet | ResNet-50 | DivNet |
|---------------------|-----|-----------|-----------|-----------|------------|------------|------------|-----------|
| Süre/İterasyon(sn.) | - | 4 | 2.6 | 2.6 | 12 | 9 | 20 | 3 |
| Toplam Parametre | - | 6.361.562 | 1.980.324 | 1.980.324 | 24.592.045 | 15.199.949 | 23.866.253 | 4.814.000 |
| Konv. Parametre | - | - | 371.712 | 371.712 | 21.802.784 | 14.714.688 | 23.587.712 | 1.614.091 |
| Toplam Süre (sn.) | 99 | 943 | 463 | 747 | 3360 | 2520 | 5600 | 840 |

ortaya çıkmaktadır. Dolayısıyla AE'nin eğitimi CAE'ye göre daha uzun sürmektedir. Burada raporlanan süre/iterasyona ek olarak AE ve CAE mimarilerinin uyguladığı gözetimsiz ön-eğitimin süresi de toplam süreye dahil edilmiştir. CAE ve CNN için filtre birleştirme tekniği oldukça yüksek hesaplama kaynağına ihtiyaç duymaktadır. Filtre birleştirme olmaksızın CAE ve CNN'in iterasyon başına süreleri sırasıyla 0.9 ve 0.5 saniyeye inmektedir. Aktarımlı öğrenme teknikleri kıyaslandığında DivNet modeli ön plana çıkmaktadır. DivNet tüm aktarımlı öğrenme yöntemleri içinde en az parametre ile en yüksek başarıyı veren model olmaktadır.





6. SONUÇLAR

Bu tez çalışmasının katkısı 3 boyutta değerlendirilebilir. Öncelikle, el işçiliği ile çıkarılan özneteliklerin, öznetelikleri otomatik öğrenen DNN'lere göre düşük başarımla verildiği kanıtlanmıştır. İkincil olarak, DNN ağırlıklarının başlatılması için literatürde önerilmiş iki yöntem olan gözetimsiz ön-egitim ve aktarımlı öğretim teknikleri derinlemesine kıyaslanmıştır. Son olarak, aktarımlı öğretim için simülasyon verileri ile bir veri tabanı oluşturulup, bu veri tabanı ile bir DNN modeli eğitildiğinde, bu modelden aktarımlı öğrenme yapmanın diğer tüm yöntemlerden üstün performans verdiğini gösterilmiştir.

İlk katkı olan geleneksel yöntemlerle DNN modellerinin kıyaslanması konusunda, hem lokal bölgelerden öznetelikler çıkartan hem de gözetimsiz ön-öğretim kullanan CAE modelinin geleneksel yöntemleri ve diğer DNN modellerini sınıflandırma performansı açısından geride bıraktığı gösterilmiştir.

İkinci katkı olan DNN ağırlık başlatma tekniklerinin kıyaslanması durumunda iki algoritma ele alınmıştır: Gözetimsiz ön-egitim ve aktarımlı öğrenme. Sonuçlar incelendiğinde, iki yöntemin de rastgele başlatılan CNN'lere göre üstün başarı sergilediği görülmüştür. Çok az sayıda eğitim verisi olduğu durumda aktarımlı öğrenmenin, rastgele başlatıma göre %25, gözetimsiz ön-öğretim kullanan CAE'ye göre ise %10 performans artışı sağladığı gözlemlenmiştir. Fakat, aktarımlı öğrenme yapan modellerin (VGGNet ve GoogleNet) mikro-Doppler ile alakasız bir veri seti üzerinden eğitilmiş olması bu modellerin elde edebileceği performansını sınırlamakta ve bu da eğitim verisi sayısının artırıldığı durumda CAE'nin performansının aktarımlı öğrenmeyi aşmasına sebep olmaktadır.

Son katkı olarak tez kapsamında 7 insan hareketini içeren bir 'çeşitlendirilmiş' mikro-Doppler simülasyon veri seti oluşturulmuştur. Bu veri seti, insan hareketlerindeki hız, boydan ve kişisel hareket stiline kaynaklı farklılıkları modelleyen 32000 veriden oluşmaktadır. Bu veri seti üzerine eğitilmiş 30 katmanlı bir artçıl öğrenme modelinden, gerçek verilerin sınıflandırılması için aktarımlı öğrenme yapılması halinde, ImageNet üzerine eğitilmiş modellerden aktarımlı öğrenme yapılmasına kıyasla oldukça yüksek başarımla alındığı gözlemlenmiştir. Son olarak, gürültü altında da önerilen yöntemin diğer yöntemlere göre gürbüz olduğu gösterilmiştir.

ÖZGEÇMİŞ

Ad-Soyad : Mehmet Saygın Seyfiođlu

Uyruđu : T.C.

Dođum Tarihi ve Yeri : 19.01.1991 - Ankara

E-posta : msseyfioglu@etu.edu.tr

ÖĐRENİM DURUMU:

- **Lisans** : 2015, TOBB ETU, Elektrik ve Elektronik Mühendisliđi

MESLEKİ DENEYİM VE ÖDÜLLER:

| Yıl | Yer | Görev |
|-----------|--|--------------------------------|
| 2016- | STM | Veri Bilimci |
| 2015-2016 | TOBB ETU | Burslu Yüksek Lisans Öğrencisi |
| 2015-2016 | TOBB ETU Radar Sistemleri Laboratuvarı | Stajyer |
| 2014-2015 | Tübitak Uzay | Stajyer |
| 2013-2014 | Tübitak Uzay | Stajyer |

YABANCI DİL: İNGİLİZCE

TEZDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **M. S. Seyfiöđlu**, and Sevgi Zübeyde Gürbüz. "Deep Neural Network Initialization Methods for Micro-Doppler Classification With Low Training Sample Support." IEEE Geoscience and Remote Sensing Letters (2017).
- **M. S. Seyfiöđlu**, A. Serinöz, A. M. Özbayođlu, S. Z. Gürbüz, "Feature Diverse Hierarchical Classification of Human Gait with CW Radar for Assisted Living," 2017 IET International Conference on Radar Systems, Belfast.
- **M. S. Seyfiöđlu**, S. Z. Gürbüz, A. M. Özbayođlu and M. Yüksel, "Deep learning of microDoppler features for aided and unaided gait recognition," 2017 IEEE Radar Conference (RadarConf), Seattle, WA, USA, 2017, pp. 1125-1130. doi:10.1109/RADAR.2017.7944373