

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PASİF WI-FI RADAR VERİLERİ İLE AKTİVİTE
SINIFLANDIRMA VE UÇTA UYGULAMASI

Sabit GÖLCÜK

YÜKSEK LİSANS TEZİ
Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Haberleşme Programı

Danışman
Doç. Dr. Nihan KAHRAMAN

Ağustos, 2024

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

PASİF WI-FI RADAR VERİLERİ İLE AKTİVİTE
SINIFLANDIRMA VE UÇTA UYGULAMASI

Sabit GÖLCÜK tarafından hazırlanan tez çalışması 06.08.2024 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Anabilim Dalı Haberleşme Programı **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Nihan KAHRAMAN
Yıldız Teknik Üniversitesi
Danışman

Jüri Üyeleri

Doç. Dr. Nihan KAHRAMAN, Danışman
Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Zehra Gülru ÇAM TAŞKIRAN, Üye
Yıldız Teknik Üniversitesi

Dr. Öğr. Üyesi Surhan BOZKURT, Üye
Doğuş Üniversitesi

Danışmanım Doç. Dr. Nihan KAHRAMAN sorumluluğunda tarafımca hazırlanan Pasif Wi-Fi Radar Verileri ile Aktivite Sınıflandırma ve Uçta Uygulaması başlıklı çalışmada veri toplama ve veri kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Sabit GÖLCÜK

İmza

*Aileme
ve
Yol Arkadaşıma*



TEŐEKKÜR

Yüksek lisans eğitimin süresinde bana destek veren, ders aldığım ve bilgi alışverişinde bulunduğum herkese, araştırma sürecim boyunca bana gösterdiği rehberlik, sabır ve değerli katkılarından dolayı, tez danışmanım Sayın Doç. Dr. Nihan Kahraman'a sonsuz teşekkürlerimi sunarım.

Geçmişten bugüne, her alanda bana manevi destek sağlayan anne, babama ve kardeşime her zaman yanımda oldukları ve bana inandıkları için minnettarım. Ayrıca bu sürecin tüm zorluklarını benimle paylaşarak her daim yanımda olan en büyük destekçim FİLİZ'e en içten teşekkürlerimi sunarım.

Sabit GÖLCÜK

İÇİNDEKİLER

SİMGE LİSTESİ	vii
KISALTIMA LİSTESİ	viii
ŞEKİL LİSTESİ	ix
TABLO LİSTESİ	x
ÖZET	xi
ABSTRACT	xiii
1 GİRİŞ	1
1.1 Literatür Taraması	1
1.2 Tezin Amacı	4
1.3 Hipotez	4
2 PASİF Wi-Fi VERİLERİ İLE İNSAN AKTİVİTE TANIMA	5
2.1 Önerilen Yöntem	5
2.2 Aktivite Tanımda Pasif Wi-Fi Radar Kullanımı	6
2.2.1 Pasif Wi-Fi Radar Sistemleri	6
2.2.2 OperaNet PWR Veri Setinin oluşturulması	8
2.3 Derin öğrenme ile Aktivite Sınıflandırma	10
2.3.1 AlexNet mimarisi ile Sınıflandırma	14
2.3.2 VGG16 mimarisi ile Sınıflandırma	15
2.3.3 Artık Ağ Mimarisi ile Sınıflandırma	16
2.3.4 Derin Öğrenme Mimarilerinde Kayıp Fonksiyonu ve Eğitim Süreci	19
2.4 Veri Ön-işleme ve İyileştirme Yaklaşımı	20
2.5 Aktivite sınıflandırma için uçta uygulama	25
3 PERFORMANS TESTLERİ	27
3.1 Hiper-parametreler ve Performans Metrikleri	27
3.2 Derin Öğrenme Modellerinin Performansa Etkisi	29

3.3 Yöntem Parametrelerinin Performansa Etkisi	33
4 SONUÇ	37
KAYNAKÇA	38
TEZDEN ÜRETİLMİŞ YAYINLAR	42



SİMGE LİSTESİ

r_a	Aktivite Sınıfları için Güven Eşik Deęeri
r_{ay}	Aktivite Yok Sınıfı için Güven Eşik Deęeri
s	Bölüt Kaydırma Parametresi
DS_i	i.Kanal için Doppler Spektrogramı
a	Eđitim Verisi Oranı
w	Spektrogram Bölüt Uzunluęu

KISALTMA LİSTESİ

AP	Erişim Noktası - <i>Access Point</i>
BN	Yığın Normalleştirme - <i>Batch Normalization</i>
CNN	Evrışimli Sinir Ağları - <i>Convolutional Neural Network</i>
ConViT	Evrışimli Görsel Dönüştürücü - <i>Convolutional Vision Transformer</i>
CSI	Kanal Durum Bilgisi - <i>Channel State Information</i>
GS	Giyilebilir Sensörler
FC	Tam Bağlantı - <i>Fully Connected</i>
FPS	Saniye başına kare - <i>Frame Per Second</i>
FT	Füzyon Dönüştürücü - <i>Fusion Transformer</i>
HAR	İnsan Aktivite Tanıma - <i>Human Activity Recognition</i>
mAP	Ortalama Hassasiyet - <i>Mean Average Precision</i>
ML	Makina Öğrenmesi - <i>Machine Learning</i>
OS	Ortam Sensörleri
PWR	Pasif Wi-Fi Radar - <i>Passive Wi-Fi Radar</i>
ResNet	Artık Ağ - <i>Residual Network</i>
RGB	Kırmızı Yeşil Mavi - <i>Red Green Blue</i>
RGB-D	Kırmızı Yeşil Mavi Derinlik - <i>Red Green Blue Depth</i>
ReLU	Doğrultulmuş Doğrusal Birimler - <i>Rectified Linear Unit</i>
SVM	Destek Vektör Makinaları - <i>Support Vector Machine</i>
SIFT	Ölçekle Değişmeyen Özellik Dönüşümü - <i>Scale Invariant Feature Transform</i>
VGG	Görsel Geometri Grubu - <i>Visual Geometry Group</i>
ViT	Görsel Dönüştürücü - <i>Vision Transformer</i>
VK	Video Kamera

ŞEKİL LİSTESİ

Şekil 2.1	PWR ile aktivite sınıflandırma için önerilen genel akış diagramı	6
Şekil 2.2	Pasif Wi-Fi veri toplama düzeneği	8
Şekil 2.3	(a) Birinci deney düzeneği, (b) İkinci deney düzeneği	9
Şekil 2.4	Deney 18 için PWR doppler spektrogramları (a) GK2, (b) GK3, (c) GK4	11
Şekil 2.5	N filtreden oluşan bir evrişim katmanının genel işleyiş adımları	12
Şekil 2.6	4x4'lük bir görüntü için max ve ortalama havuzlama işlemleri	13
Şekil 2.7	CNN eğitim ve test aşaması	14
Şekil 2.8	(a) ResNet18 mimarisi (b) Soldan sağa, Birinci ana temel blok, birim bağlantılı BON (B1-ID), Üçüncü ana temel blok, izdüşüm bağlantılı BON (B3)	17
Şekil 2.9	Her bir deney için, spektrogramların eğitim ve test verisine dönüştürülmesi	21
Şekil 2.10	Deney 18 için 3 kanallı pwr doppler spektrogramı ve farklı aktive sınıfları için spektrogram verisinin değişimi	21
Şekil 2.11	3B spektrogramın w ve r parametrelerine göre bölütlenmesi ($f(w, r)$)	23
Şekil 2.12	Farklı sınıflar için spektrogram bölütleri (a) Aktivite Yok, (b) Yere Uzanma, (c) Ayakta Durma (d) Dönme	25
Şekil 3.1	AlexNet karmaşıklık matrisi	31
Şekil 3.2	VGG16 karmaşıklık matrisi	31
Şekil 3.3	ResNet18 karmaşıklık matrisi	32
Şekil 3.4	ResNet34 karmaşıklık matrisi	32
Şekil 3.5	Önerilen yöntem ve Durum-5 için sınıf bazlı doğruluk değerleri	34

TABLO LİSTESİ

Tablo 1.1	Farklı aktivite tanıma çalışmaları için, sensör ve veri tiplerinin karşılaştırılması, GS: Giyilebilir sensörler, OS: Ortam Sensörleri, VK: Video Kamera	2
Tablo 2.1	Deney senaryoları ve içerdikleri aktivite sınıfları (AY: Aktivite Yok)	9
Tablo 2.2	Gerçekleştirilen aktivite süreleri	10
Tablo 2.3	AlexNet sınıflandırıcı mimari ve parametreleri	15
Tablo 2.4	VGG16 sınıflandırıcı mimari ve parametreleri	16
Tablo 2.5	ResNet18 sınıflandırıcı mimari ve parametreleri	18
Tablo 2.6	Raspberry Pi 5 donanım özellikleri	26
Tablo 3.1	Derin öğrenme modelleri hiper-parametreleri	28
Tablo 3.2	Farklı omurga ağları ve literatür yöntemleri için GFlops, parametre sayısı, doğruluk ve F1-ölçütleri	30
Tablo 3.3	Parametrelerin sınıflandırma performansına ve eğitim seti boyutuna etkisi, (ResNet18, $w = 4$). Durum-1, önerilen yöntemdir	33
Tablo 3.4	Durum-1 ve Durum-5 için eğitim ve test veri seti aktivite sınıflarına ait örnek sayıları	34
Tablo 3.5	Farklı omurga ağları ile için, doğruluk ve F1-ölçütünün iki farklı parametre setinde karşılaştırılması	35
Tablo 3.6	Raspberry Pi 5 ile farklı derin öğrenme mimarileri için hız karşılaştırması	36

Pasif Wi-Fi Radar Verileri ile Aktivite Sınıflandırma ve Uçta Uygulaması

Sabit GÖLCÜK

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Doç. Dr. Nihan KAHRAMAN

Aktivite sınıflandırma günümüzde, akıllı ev sistemleri, bakım gerektiren kişilerin hareketlerinin otomatik olarak takip edilmesi, davranış analizi gibi birçok uygulama için oldukça önemlidir. Geliştirilen bir uygulamanın, doğrudan yerel cihazlar üzerinde çalıştırılmasına olanak tanıyan uçta uygulama (edge computing), aktivite sınıflandırma projelerinde de oldukça önemli bir rol oynar. Bunun nedeni gerçek zamanlı uygulamalarda hız ve verimlilik sağlamanın yanı sıra, gizlilik ve veri güvenliği konularında bulut tabanlı sistemlere göre avantajlar sağlamasıdır. Bu çalışmada, düzenli olarak iletilen Wi-Fi sinyallerinin, pasif Wi-Fi radar (PWR) sistemleri ile yakalanarak frekans kaymalarının zaman içinde nasıl değiştiğini gösteren spektrogram görüntüleri ile aktivite sınıflandırma modeli önerilmiştir. Sonrasında geliştirilen uygulamanın Raspberry Pi üzerindeki performansı incelenmiş ve önerilen sistemin hem hızlı hem de yüksek doğruluk oranında çalışabildiği gösterilmiştir. Literatürden farklı olarak, önerilen veri iyileştirme ve kapsamlı ön işleme teknikleri ile eğitim veri setinin kalitesi iyileştirilerek model tarafından daha etkin bir şekilde işlenmesi sağlanmıştır. Böylece sınıflandırma doğruluğu artırılarak, ilgili veri tabanında en yüksek başarıma ulaşan bir model oluşturulmuştur.

Çalışmada, farklı karmaşıklıkta sınıflandırıcılar kullanılarak uçtan uca eğitim yapılmış ve önerilen veri iyileştirme tekniklerinin farklı mimarilerdeki performansı karşılaştırılmıştır. Eğitim ve test verisi olarak 6 farklı aktivite sınıfı (oturma, kalkma, yürüme, yere uzanma, dönme, yerden kalkma) ve 1 aktivite olmama durumunu içeren OperaNet veri seti kullanılmıştır. Önerilen model eğitimleri

Google Colab ile NVIDIA T4 GPU kullanarak gereklenmiřtir. Uta uygulama iin test ařamasında Raspberry Pi 5 platformu kullanılmıřtır.

Sonular, nerilen veri iřleme ve iyileřtirme teknikleri ile, referans ynteme kıyasla sınıflandırma performansının %8 oranında iyileřtirdiđini gstermektedir. Raspberry Pi zerinde gerekleřtirilen testlerde ise, AlexNet 15 fps, VGG16 1.2 fps, ResNet18 8 fps, ResNet34 5 fps hız ile alıřtıđı raporlanmıřtır. Bu tez kapsamında kullanılan tm sınıflandırıcılara ait model dosyaları <https://github.com/sgolcuk/Activity-Recognition-With-PWR> adresinde aık eriřime sunulmuřtur.

Anahtar Kelimeler: Aktivite tanıma, veri n iřleme, pasif Wi-Fi radar, uta uygulama



Activity Recognition Using Passive Wi-Fi Radar Data and Edge Implementation

Sabit GÖLCÜK

Department of Electronics and Communications Engineering
Master of Science Thesis

Supervisor: Assoc. Prof. Nihan KAHRAMAN

Nowadays, activity classification is very important for many applications such as smart home systems, automatic tracking of the movements of people requiring care, behavior analysis. Edge computing, which allows an application to be run directly on local devices, plays an important role in activity classification projects. This is because it provides speed and efficiency in real-time applications as well as advantages over cloud-based systems in terms of privacy and data security. In this study, an activity classification model is proposed by capturing regularly transmitted Wi-Fi signals with passive Wi-Fi radar (PWR) systems and using spectrogram images that show how frequency shifts change over time. Then, the performance of the developed application on Raspberry Pi is analyzed and it is shown that the proposed system is both fast and highly accurate. Unlike the literature, the quality of the training dataset is improved, with the proposed data augmentation and extensive preprocessing techniques, in that way processed more efficiently by the model. Thus, a model that achieves the highest performance on the relevant database has been created by increasing the classification accuracy.

In this study, end-to-end training is performed using classifiers of different complexity and the performance of the proposed data enhancement techniques on different architectures is compared. OperaNet dataset containing 6 different activity classes (sit, stand, walk, lie-down, body-rotate, stand-from-lie) and 1 non-activity case is used as training and test data. The proposed model trainings were implemented with Google Colab using NVIDIA T4 GPU. Raspberry Pi 5 platform was used in the testing phase for the application at the edge.

The results show that the proposed data processing techniques improve the classification performance by 8% compared to the reference method. The tests performed on the Raspberry Pi were reported to run at AlexNet 15 fps, VGG16 1.2f ps, ResNet18 8 fps, and ResNet34 5 fps. The model files of all classifiers used in this thesis are available for open access at <https://github.com/sgolcuk/Activity-Recognition-With-PWR>.

Keywords: Activity recognition, data pre-processing, passive Wi-Fi radar, edge computing



1.1 Literatür Taraması

İnsan Aktivite Tanıma (Human Activity Recognition - HAR), insan davranışlarının ve hareketlerinin belirlenip sınıflandırılmasını sağlayan bir çalışma alanıdır. Bu sistemler, uygulama ortamlarına göre kapalı alan (indoor) ve açık alan (outdoor) insan izleme sistemleri olarak sınıflandırılabilir ve insan-bilgisayar etkileşimi, sağlık amaçlı izleme, akıllı ev ve gözetleme sistemleri gibi çeşitli alanlarda uygulanabilir [1–5]. Mevcut HAR sistemleri, çevreden veri toplamak ve aktiviteleri tanımak için RGB video kameralar [6–9], Kinect gibi derinlik algılama teknolojileri [10, 11] ve giyilebilir sensörler [2, 12] gibi farklı sensörleri tek başlarına veya birlikte kullanır [13–16]. Tablo 1.1’de farklı sensör ve verileri girdi olarak alan ve aktivite tanıma yapan yöntemlere örnekler, kullanılan veri setleri ve sınıf sayılarıyla birlikte karşılaştırmalı olarak verilmiştir.

Görsel tabanlı HAR, insan hareketlerini içeren görüntüleri analiz etmektedir. Burada zamansal bilgiyi dikkate almadan tek bir görüntüden insan aktivitelerini tanıyan yöntemler olduğu gibi, video kareleri arasındaki hareket bilgisini kullanan farklı yaklaşımlar da vardır. D. Girish vd. [17], tek bir görüntü üzerinden aktivite tanıma için, görüntüyü YoloV2 nesne sezici ile farklı bölgelere ayırmakta ve insan ve obje olarak belirlenen her bir bölgenin tekil, birleşim ve kesişim bölgelerinden AlexNet omurga mimarisi ile öznelikler elde edilmektedir. Bu özelliklerin farklı kombinasyonları, Destek Vektör Makineleri (SVM) ve lojistik regresyon sınıflandırıcılar ile sınıflandırılmaktadır. Farklı veri setlerinin birleşiminden oluşturulmuş, 8 farklı aktivite sınıfı içeren yeni bir veri setinde %81,97’lik bir sınıflandırma doğruluğu elde edilmiştir. [18]’de görüntü içindeki farklı bölgeler arasındaki ilişkiler, Vision Transformer (ViT) ile çıkartılır. Bu amaçla, tek görüntüden ResNet50(Artık Ağ-Residual Network) ile elde edilen öznelikler iki adet ViT’den geçirilerek, aktivite tanıma gerçekleşir. ConViT olarak adlandırılan bu model ile, Stanford40 veri setinde yapılan deneylerde %95,5 mAP (mean Average Precision) oranına ulaşılarak geleneksel yöntemlere kıyasla

doğruluğun artırıldığı gösterilmiştir. Hareketsiz görüntülerin hareketin yönü, hız ve zaman içindeki değişim gibi kritik bilgileri içermemesinin ortaya çıkardığı zorluk nedeniyle, zamansal bilgiyi kullanarak sınıflandırma yapan birçok yöntem önerilmiştir. [7]'de aktivite tanıma için zamansal bilgi, video kareler arası hesaplanan Ölçekle Değişmeyen Özellik Dönüşümü (SIFT-Scale Invariant Feature Transform) akış vektörleri kullanılarak elde edilmekte, hareket ve renk özellikleri birlikte kullanılarak performans artırılmaktadır.

Tablo 1.1 Farklı aktivite tanıma çalışmaları için, sensör ve veri tiplerinin karşılaştırılması, GS: Giyilebilir sensörler, OS: Ortam Sensörleri, VK: Video Kamera

Yöntem	Sensör	Veri	Veri Seti (# sınıf)
[7]	VK	Optik Akış vektörleri	UCF 101 (101)
[9]	VK	İskelet verisi	UTD-MHAD (27) KTH (6) UCF-sports (10)
[10]	Kinect	İskelet verisi	UI-PRMD (10) KIMORE (5) EHE (6)
[12]	GS	İvmeölçer, jiroskop, manyetometre sinyalleri	Opportunity (18) PAMAP2 (12) UniMiB-SHAR (17)
[13]	VK Kinect	RGB görüntü, iskelet verisi	NTU RGB+D (60,120) PKU-MMD (51) N-UCLA (12) Toyota Smarthome (31)
[15]	VK GS OS	RGB görüntü İvmeölçer, jiroskop sinyalleri EEG sinyalleri Parlaklık sensör verileri Kızılötesi sensör verileri	UP-Fall (11)

RGB videodan elde edilen uzamsal-zamansal bilginin performansı arttırdığı birçok çalışmada gösterilmekle birlikte, gerçek video görüntülerinde ortam gürültüsü, ışıklılık değişimleri gibi zorluklara karşı gürbüzlük sınırlıdır. Son yıllarda yapılan çalışmalarda, RGB-D kamera, Kinect gibi sensörlerinden elde edilen derinlik bilgisinin kullanılmasının performansı arttırdığı gösterilmiştir. [19]'de RGB-D sensöründen elde edilen, renkli görüntü, ön, arka ve yukarı bakış açıları ile elde edilen derinlik bilgisi ve iskelet verisinin 5 farklı VGG16 sınıflandırıcı ile sınıflandırıldığı bir model önerilmiştir. Her bir sınıflandırıcıdan elde edilen skor değerlerinin çarpılması sonucu elde edilen son skor vektörünü en büyükleyen değer belirttiği sınıf aktivite sınıfı olarak belirlenmektedir. [14], RGB-D sensörden elde edilen iskelet verisini, RGB görüntüsü ile birleştirerek önemli kareleri belirlenmek amacıyla kullanır. Belirlenen önemli kareler uzamsal-zamansal bilginin çıkarımı için önerilmiş özgün bir modüle girdi olarak

verilir. Bu çıktı ve iskelet verisini girdi olarak alan graf konvolüsyonel ağı çıktıları, karar tümleştirme ile birleştirilir ve aktivite tanıma gerçekleşir.

Görüntü (video) sensörlerinden farklı olarak, kişi üzerinde taşınabilen ve vücut hareketlerini algılayabilen ivmeölçerler, jiroskoplar gibi giyilebilir cihazlar, HAR uygulamalarında sıklıkla kullanılan diğer sensörlerdendir [2, 11, 12] . [12], vücudun farklı noktalarına yerleştirilmiş üç eksenli ivmeölçer, jiroskop ve manyetometre sensörlerinden elde edilen verileri kullanarak günlük aktiviteleri tanıma derin öğrenme tabanlı bir yöntem önermiştir. Benzer şekilde ivmeölçer ve jiroskopdan elde edilen bilgileri kullanan [2]'de ise veriler, mobil telefonlarda gömülü olarak yer alan sensörlerden elde edilmiştir.

Yukarıda açıklanan sensör teknolojilerin kullanımı bazı önemli zorluklara neden olabilmektedir. Örneğin, giyilebilir cihazların pil ömrüne olan bağımlılığı ve kullanıcı rahatsızlığı, RGB videoların ise mahremiyet ihlalleri ve aydınlatma koşullarına olan duyarlılığı gibi sorunlar bulunmaktadır [20]. HAR sistemlerinde ayrıca pasif algılama sensörleri ve Wi-Fi tabanlı ortam sensörleri gibi alternatif teknolojiler de kullanılmaktadır [21–25]. Bu sensörler, kişilerle doğrudan fiziksel bir etkileşimde bulunmadan, insanların bulunduğu ortamdaki sinyal değişikliklerini, özellikle Wi-Fi sinyallerinde görülen frekans kaymalarını ve zaman gecikmelerini takip ederek aktiviteleri tespit eder. Bu bağlamda, pasif Wi-Fi radar (PWR) ve kanal durumu bilgisi (Channel State Information - CSI), yaygın olarak kullanılmaktadır. PWR, ortamdaki mevcut sinyallerinin çevredeki nesnelere yansımalarını ve bu yansımaların değişimini analiz ederken; CSI, gönderici ile alıcı arasındaki kanal üzerinde sinyalde oluşan genlik zayıflaması ve faz kayması gibi özellikleri ölçer. [25]'de, çeşitli alıcı sayısına sahip CSI ve PWR verilerinden spektrogram, skalogram ve Markov geçiş alanı (Markov transition field) ile görsel-tabanlı veriler oluşturulur. Farklı sensörlerden elde edilmiş bu görüntüler, önerilen bir görsel (vision) transformer mimarisi ile tümleştirilerek aktivite sınıfı belirlenir. Yöntem OperaNet CSI+PWR veri setinde [26] 15 farklı özneliği birleştirerek %92.8'lik bir doğruluğa ulaşmıştır. WiWeHAR'da [27] CSI ve giyilebilir sensör bilgileri tümleştirilerek, ayırık kullanımlarına göre daha yüksek performansa ulaşıldığı gösterilmiştir. Bundan farklı olarak [28], CSI ve RGB görüntüler ile ayırık aktivite tanıma yapar ve karar düzeyinde tümleştirme gerçekleştirir. Her iki yöntem için de performans, kendi oluşturdukları veri setlerinde raporlanmıştır.

1.2 Tezin Amacı

Derin öğrenme modellerinde, veri kalitesi ve etkili bir ön işleme süreci, modelin başarısını ve genelleme kabiliyetini önemli ölçüde artırır. Ayrıca, veri iyileştirme yöntemleri kullanarak yapılan çeşitlendirme (augmentation), modelin aşırı uyum (overfitting) problemini önlemeye yardımcı olur. Bu sayede, modelin farklı senaryolarda daha gürbüz ve doğru çalışması amaçlanır. Bu çalışmada, veri ön işleme aşamasında, giriş verileri yeniden yapılandırılarak ve zenginleştirilerek veri kalitesi artırılmıştır. Bu süreç, verilerin derin öğrenme modelleri tarafından daha etkin bir şekilde işlenmesini sağlamış ve sınıflandırma doğruluğunu önemli ölçüde iyileştirmiştir. Özellikle, veri seti içindeki bazı aktivite sınıfları için performans artırılmıştır. Önerilen veri ön-işleme ve iyileştirme yöntemlerinin performansının incelenmesi amacıyla, Pasif Wi-Fi Radar (PWR) sistemleri ile elde edilen veriler kullanılmış ve aktivite tanıma gerçekleştirilmiştir. Ayrıca, önerilen yaklaşımın farklı derin öğrenme mimarilerdeki performansa etkisi de irdelenmiştir. Bu amaçla, literatürde sıklıkla kullanılan AlexNet[29], VGG16[30], ResNet[31] evrişimli sinir ağları ile eğitim gerçekleştirilerek, karşılaştırmalı performans raporlaması yapılmıştır. Test aşamasında, modelin gerçek uygulama koşullarında ve görece daha az güçlü bilgisayarlarda nasıl performans gösterdiğini değerlendirmek için, geliştirilen tüm modeller modüler bir bilgisayar olan Raspberry Pi 5 cihazında koşulmuş, doğruluk ve hız karşılaştırmalı olarak raporlanmıştır.

1.3 Hipotez

Derin öğrenme modellerinin veriyi genelleme yeteneği ve performansı, eğitim ve test verisinin kalitesi ile doğrudan ilişkilidir. Bu çalışmada, önerilen yenilikçi ön işleme yaklaşımları ile, farklı makine öğrenmesi modellerinin eğitim ve test süreçlerinde veri kalitesinin ve dolayısıyla aktivite sınıflandırma başarısının artırılacağı hipotezi öne sürülmektedir. Ayrıca, bu modellerin Raspberry Pi gibi görece daha az güçlü bilgisayarlar üzerinde uçta uygulanabilirliği gösterilmiştir. Önerilen yüksek performanslı aktivite tanıma modelinin, gerçek zamanlı çalışabilen, düşük maliyetli ve taşınabilir bu sisteme uyumluluğu sayesinde gerçek dünya uygulamaları için uygun bir alternatif sağlayacağı düşünülmektedir.

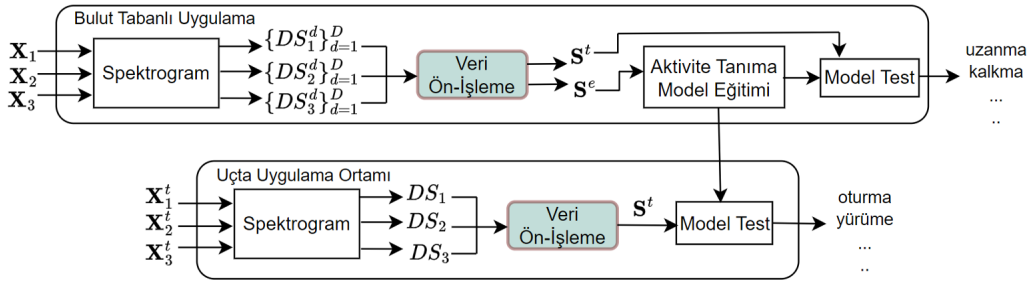
2

PASİF Wi-Fi VERİLERİ İLE İNSAN AKTİVİTE TANIMA

2.1 Önerilen Yöntem

Bu tez kapsamında, pasif Wi-Fi sensörlerinden elde edilen verileri kullanarak insan aktivitelerinin tanıma ve sınıflandırılması için yeni bir veri iyileştirme ve zenginleştirme yöntemi önerilmekte ve önerilen yöntemlerin farklı derin öğrenme mimarilerinde aktivite sınıflandırma performansına etkisi incelenmektedir. Özel olarak tasarlanmış bir ortamda yer alan alıcı ve verici sensörler ile toplanan Wi-Fi sinyalleri, insan hareketlerinden kaynaklanan frekans kaymaları ve zaman gecikmeleri içerir. Önerilen yaklaşım, spektrograma dönüştürülmüş bu verilerden derin öğrenme modelleri kullanarak özellik çıkarma ve sınıflandırma işlemleri gerçekleştirmek üzerine kurulmuştur. Şekil 2.1’de bu tez çalışmasında önerilen bulut tabanlı ve uçta uygulama modellerinin akış diagramı görülmektedir. Model eğitimlerinin gerçekleştirilmesi amacıyla, öncelikli olarak, pasif Wi-Fi radar (PWR) sistemindeki her bir alıcının (receiver) yakaladığı Wi-Fi sinyalleri (\mathbf{X}_i , $i=1,2,3$), Doppler frekans analizi ile işlenerek zaman eksenini boyunca frekans değişimlerini gösteren spektrogramlara ($\{DS^d_i\}_{d=1}^D$, $i=1,2,3$) dönüştürülür. Burada d , veri toplama aşamasında veri çeşitliliğini arttırarak model genelleme performansını iyileştirmek amacıyla oluşturulmuş deney senaryolarını, D ise toplam deney sayısını ifade etmektedir. Elde edilen Doppler spektrogramları, aktivite tanıma model performansını artırmak amacıyla bir dizi kapsamlı ön işleme adımlarından geçirilir. İlk olarak, 3 kanallı spektrogram görüntülerinden bölütler oluşturulur. Bu işlem veri sayısı ve çeşitliliğinin artırılmasına olanak tanımaktadır. Bölütlenmiş verilerin etiketlenmesi sürecinde ise, önerilen güven eşikleme yöntemi uygulanır. Bu işlem, verinin kalitesini ve doğruluğunu artırmak için, anlamlı veri bölütlerinin seçilmesini sağlayarak, eğitim aşamasında daha güvenilir sonuçlar elde edilmesine olanak verir. Kullanılan eğitim (S^e) ve test verileri (S^t) etiketlenmiş bölütlerden rastgele ve sınıflar arası denge gözetilerek oluşturulmaktadır. Makine öğrenmesi modelinin eğitilmesi sonrasında, test

verileri üzerinde performansı değerlendirilir. Önerilen yöntemin gerçek dünya uygulamalarında kullanılabilirliğinin gözlemlenmesi amacıyla, test aşaması bulut tabanlı ortamın yanı sıra uçta uygulama ortamında da gerçekleştirilmiştir. Burada anlık olarak toplanan verilerin spektrograma dönüştürülmesi ve benzer ön işleme adımlarından geçirilmesinin ardından model testleri gerçekleştirilmiş, performans ve hız gibi farklı metrikler üzerinden karşılaştırmalar yapılmıştır. Bu tez çalışması kapsamında, ön işleme adımları ve derin öğrenme modellerinin eğitimi için Google Colab platformu üzerinde NVIDIA T4 GPU kullanılırken, uçta uygulama Raspberry Pi 5 donanımı ile gerçekleştirilmiştir. Raspberry Pi 5, kompakt yapısı ve birçok uygulama için yeterli işlem yükü sayesinde, gerçek dünya uygulamaları için uygun bir platform olarak sıklıkla tercih edilmektedir. Her iki ortamda da, derin öğrenme modellerinin eğitimi ve test aşamaları için PyTorch kütüphanesi kullanılmıştır.



Şekil 2.1 PWR ile aktivite sınıflandırma için önerilen genel akış diagramı

2.2 Aktivite Tanımadaki Pasif Wi-Fi Radar Kullanımı

Bu bölümde, Pasif Wi-Fi Radar sistemlerinden kısaca bahsedilmektedir. Sonrasında, çalışmada insan aktivite tanıma için veri ön işleme ve iyileştirme adımlarına girdi olarak verilen referans veri OperaNet [26, 32] açıklanmaktadır.

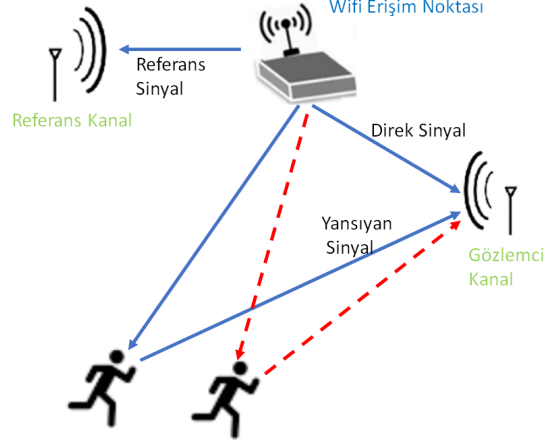
2.2.1 Pasif Wi-Fi Radar Sistemleri

Askeri, havacılık, denizcilik ve trafik izleme gibi çeşitli alanlarda kullanılan radar sistemleri, hedefleri tespit etmek, izlemek ve bunlar hakkında bilgi toplamak için elektromanyetik dalgaları kullanan sistemler olarak tanımlanmaktadır [33]. Bu teknoloji, bir verici tarafından gönderilen radyo dalgalarının bir nesneye çarptıktan sonra yansması ve bu yansıyan dalgaların bir alıcı tarafından toplanması prensibine dayanır. Yansıyan dalgaların analizi, hedefin konumu, hızı, boyutu ve diğer özellikleri hakkında bilgileri elde etmemizi sağlamaktadır. Radar sistemleri, kullanılan sinyalin türüne göre *aktif* ve *pasif* olmak üzere iki ana kategoride incelenebilir.

Aktif radar sistemleri, hedefleri tespit etmek, izlemek ve bilgi çıkarımı yapmak için kendi sinyallerini yayar ve bu sinyallerin nesnelere nasıl yansıdığını izler. Bu amaçla, verici aracılığıyla yayılan ve hedef nesneye çarpılarak yansıyan sinyaller radar sisteminin anteni tarafından alınarak alıcıya iletilir. Yansıyan sinyalin, zaman gecikmesi ve frekans kayması benzeri analizleri yapılarak hedefin uzaklığı ve hızı gibi kritik bilgileri ortaya çıkartılır.

Pasif radar sistemleri ise bundan farklı olarak, çevrede zaten mevcut olan sinyalleri kullanır. Yani, kendi sinyallerini yaymazlar, bu nedenle de daha az fark edilir ve elektromanyetik kirlilik yaratmazlar. Ayrıca hem maliyet-etkin hem de enerji açısından verimlidir. Pasif radarlar, mevcut sinyallerin nesnelere tarafından nasıl değiştirildiğini analiz ederek çalışır ve bu sayede hedefin konumunu ve diğer özelliklerini belirleyebilir. Bu sistemler, ortamdaki mevcut FM radyo, televizyon yayınları, hücresel ağ sinyalleri ve Wi-Fi gibi çeşitli elektromanyetik sinyalleri kullanarak bu sinyallerin nesnelere nasıl yansıdığını analiz eder [34]. Wi-Fi sinyalleri, özellikle iç mekanlarda yaygın olarak bulunduğu ve geniş bir kapsama alanı sağladığından, bu tür ortamlarda pasif radar uygulamaları için idealdir. Pasif Wi-Fi radar (PWR) olarak adlandırılan bu sistemler, çevresel Wi-Fi erişim noktalarından düzenli olarak yayılan sinyalleri algılar. Wi-Fi kullanımının bir diğer avantajı, ek bir donanım veya yatırım gerektirmeden mevcut altyapıyı kullanabilmesidir. Bu maliyetleri düşürür ve kurulumu kolaylaştırır. PWR sistemleri ayrıca, gizlilik konusunda hassas olan, kamera kullanımının uygun olmadığı ve giyilebilir cihazların problem yaratabileceği ortamlarda insan hareketlerini izlemek için bir alternatif olarak kullanılmaktadır.

Pasif Wi-Fi radar sistemleri, ortamda bulunan bir Wi-Fi vericisi (Erişim Noktası - Access Point-AP) yanı sıra, bir referans kanal ve bir veya daha fazla gözlemci kanal içerir. Şekil 2.2'de bir PWR sisteminin temsili görülmektedir. Bir Wi-Fi Erişim Noktasından (access point-AP) düzenli olarak iletilen sinyal, izlenen alanda bulunan hareketli nesneden (bu çalışmada insan) yansır. Sistemde bulunan referans kanal AP'den doğrudan gelen sinyali herhangi bir yansıma veya hareket etkisi olmadan alır. Gözlemci kanal ise çevredeki nesnelere yansıyan sinyalleri toplar. Ortamda bir kişi veya hareket mevcutsa, zaman içinde yansıyan sinyalde değişiklik meydana gelmektedir (Şekil 2.2 kırmızı çizgiler). Bu yansıyan sinyaller, orijinal sinyal ile karşılaştırılarak nesnenin yerini, hızını ve diğer hareket özelliklerini tespit etmek için kullanılır. Bu işlem, sinyaller arası korelasyonun hesaplanması sonrasında zaman gecikmesi ve frekans kaymalarının analiz edilmesi ile gerçekleştirilir.



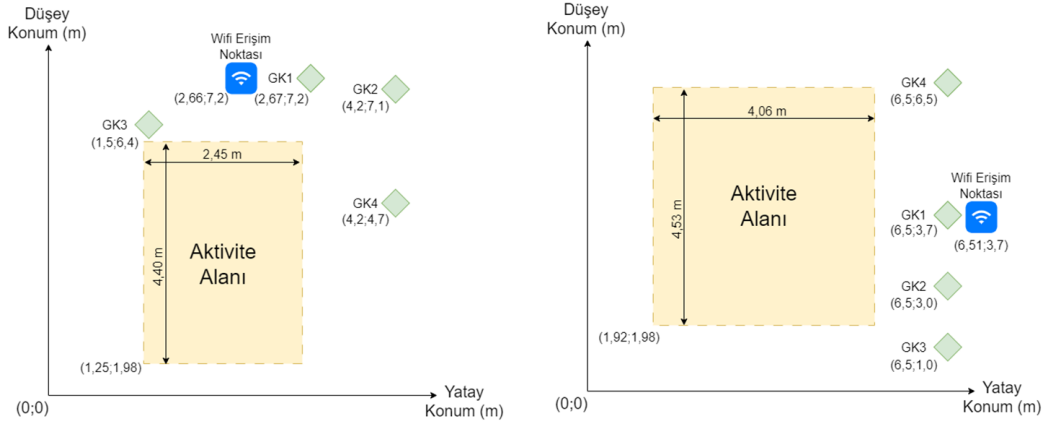
Şekil 2.2 Pasif Wi-Fi veri toplama düzeneği

Bu çalışmada kullanılan OperaNet PWR veri setinde[26, 32], birden fazla gözlemci kanal için frekans kaymalarının zaman içinde nasıl değiştiğini gösteren Doppler spektrogram değerleri raporlanmıştır. Bir sonraki bölümde veri setinin oluşturulma aşamaları ve özellikleri ayrıntılarıyla açıklanmaktadır.

2.2.2 OperaNet PWR Veri Setinin oluşturulması

OperaNet PWR veri seti, Bristol Üniversite'nde yer alan iki farklı odada yapılan deneyler ile oluşturulmuştur [26, 32]. Gerçek problemlere uygunluğu sağlamak amacıyla, odalar masa, sandalye, bilgisayar benzeri eşyalarda içermektedir. Şekil 2.3'de ilgili odalara ait planlar ve PWR verisinin elde edilmesi için gerekli olan referans ve gözlemci kanalların konumları gösterilmektedir. Sistem, 1'i referans kanal olarak kullanılan 4 gözlemci kanal (GK) içermektedir. Odaların iki farklı boyut ve düzende oluşturulması sayesinde, veri çeşitliliğinin artırılması sağlanmıştır. Şekil 2.3(a) ve Şekil 2.3(b) sırasıyla aktivite alanı (insanın aktiviteleri gerçekleştirdiği bölge) 4,4 x 2,45 metre boyutlarında olan *Oda 1* ve aktivite alanı 4,06 x 4,53 metre boyutlarında olan *Oda 2*'deki kurulum düzenini göstermektedir. Wi-Fi erişim noktası, GK1 referans kanal ve GK2, GK3, GK4 gözlemci kanallara ait konumlar grafik üzerinde belirtilmiştir. Görüldüğü gibi her iki durumda da referans kanal GK1, Wi-Fi erişim noktasına yakın konumlandırılmıştır. Bu, sinyal zayıflaması ve gürültüden etkilenmeyi en aza indirerek, güçlü ve doğru bir referans sinyal alınmasını sağlar. Böylece, gözlemciler tarafından alınan yansıyan sinyaller ile karşılaştırma hassasiyeti artırılır, küçük değişiklikler ve Doppler frekans kaymalarının daha net olarak tesbit edilmesi sağlanır. OperaNet'te, gelen sinyaldeki girişimin yol açtığı bozucu etkinin ortadan kaldırılması amacıyla CLEAN [35] algoritması kullanılmış ve Doppler spektrogramları bin değeri 200 alınarak elde edilmiştir. Bu her bir an ve gözlemci kanal için 1x200 boyutlu bir

vektörün raporlandığı anlamına gelmektedir. PWR sistemi spektrogram verilerini 10 Hz ölçüm hızında kaydetmektedir. Bu nedenle her bir saniye için, 3x10x200'lük spektrogram verisi elde edilmektedir. Burada 3, gözlemci kanal sayısını; 10, saniyede yapılan ölçüm sayısı (10 Hz) ve 200, Doppler bin sayısını ifade etmektedir.



Şekil 2.3 (a) Birinci deney düzeneği, (b) İkinci deney düzeneği

Kullanılan veri seti oluşturulurken, *yürüme*, *oturma*, *ayakta durma*, *yere uzanma*, *yerden kalma*, *dönme* olarak adlandırılan 6 aktivitenin gerçekleşmesi sırasında ölçümler gerçekleştirilmiş olup, ayrıca aktivite olmaması durumunda (*Aktivite yok - AY*) ölçümler yapılmıştır. Toplam 7 sınıftan oluşan veri seti, belirli bir aktivitenin veya aktivite dizisinin gerçekleştirildiği 36 farklı deney senaryosu içermektedir. Tablo 2.1'de 36 farklı deneyin yapıldığı oda ve deneylerin içerdiği sınıflar listelenmiştir. Görüldüğü üzere bir ya da iki aktivitenin tekrarlandığı deneyler olduğu gibi tüm aktiviteleri barındıran deneyler de bulunmaktadır. 18,29-33 nolu deneyler, deneklerin tüm aktiviteleri bir sıra belirtilmeksizin ve rastgele biçimde gerçekleştirdiği ölçümleri içerirken, 56-61 nolu deneyler, aktivitelerin önceden planlanmış bir sırada gerçekleştirildiği ölçümleri içermektedir. Deneyler, deney hakkında bilgilendirilmiş 6 katılımcı ile gerçekleştirilmiş olup her bir deney, odada yalnızca bir katılımcının bulunduğu koşullarda yürütülmüştür.

Tablo 2.1 Deney senaryoları ve içerdikleri aktivite sınıfları (AY: Aktivite Yok)

Deney No	Oda No	İçerdiği Sınıflar
deney 2,6,10,14,20,24	Oda 1	Yürüme, AY
deney 3, 7, 11, 15, 21, 25	Oda 1	Oturma, Kalkma, AY
deney 4, 8, 12, 16, 22, 26	Oda 1	Yere Uzanma, Yerden Kalkma, AY
deney 5, 9, 13, 17, 23, 27	Oda 1	Dönme, AY
deney 18, 29 - 33	Oda 1	Tüm aktiviteler
deney 56 - 61	Oda 2	Tüm aktiviteler

Tablo 2.2’de her bir aktivitenin 36 deney boyunca toplamda ne kadar süre gerçekleştiği raporlanmıştır. Görüldüğü gibi, 124 dakika ile aktivite olmama durumu verilerin büyük bir bölümünü oluşturmaktadır. Dönme ve Yürüme sınıfları en uzun süreli gerçekleştirilen aktiviteler olurken (yaklaşık 75’er dakika), yere uzanma ve yerden kalkma en kısa süreli (yaklaşık 26’şar dakika) aktivitelerdir. Süreler arasındaki bu farklılık oluşturulacak olan veri setindeki sınıf dengesi açısından önem teşkil etmekte ve sınıflandırma performansına etki etmektedir.

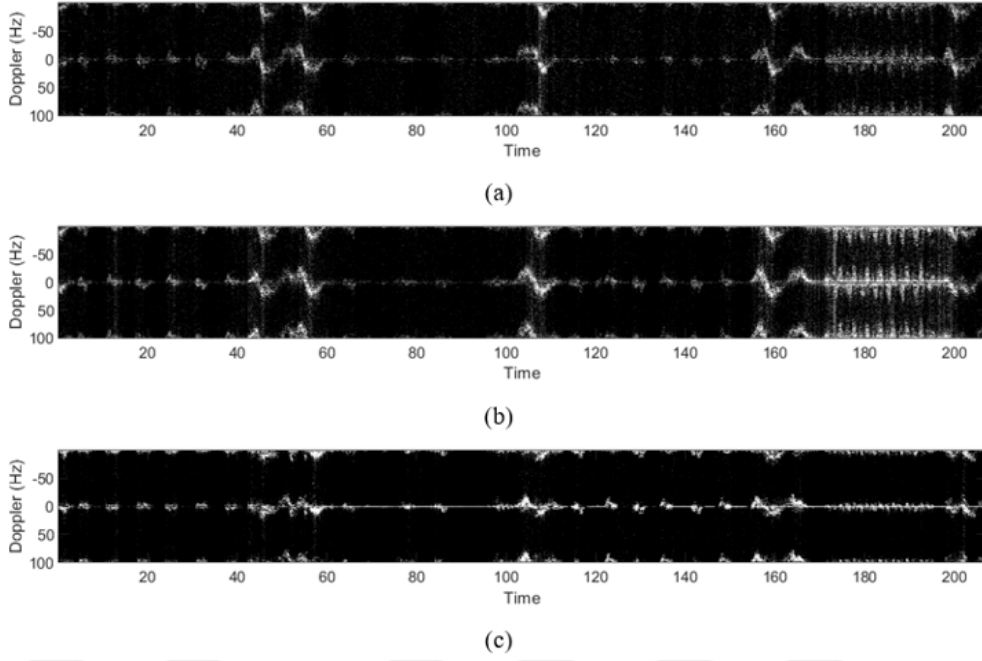
Tablo 2.2 Gerçekleştirilen aktivite süreleri

Aktivite	Süre(dk)
Yere Uzanma	26.69
Yerden Kalma	26.42
Yürüme	75.93
Oturma	35.25
Kalma	34.88
Dönme	74.59
Aktivite Yok	124.86

Bir deney için (Deney 18), kullanılan Doppler spektrogramı Şekil 2.4’de görselleştirilmiştir. Burada, (a)-(b) ve (c) sırasıyla, GK2, GK3 ve GK4 gözlemci kanallarından elde edilen Doppler spektrogramlarını göstermektedir. bu spektrogramlar farklı aktivite sınıflarında ölçümler içermekte olup , toplamda 207 saniye boyunca alınan verileri kapsamaktadır. Şekil, her bir gözlemci kanaldan elde edilen spektrogramların farklı bilgiler içerdiğini göstermesi açısından önemlidir. Belirli bir hareketin izleri bir kanalda belirgin olabilirken, başka bir kanalda daha az belirgin olabilir. Daha doğru bir sınıflandırma yapmak amacıyla, her üç kanaldan gelen bilginin bir arada değerlendirilmesi gerekliliği açıktır. Bu amaçla yapılan işlemler ve ön işleme adımları Bölüm 2.4’de ayrıntılı olarak açıklanmaktadır.

2.3 Derin öğrenme ile Aktivite Sınıflandırma

Bir denetimli öğrenme (supervised learning) yaklaşımı olan sınıflandırma, makine öğrenmesi ve özellikle derin öğrenme tekniklerinin temel uygulama alanlarından biridir. Denetimli öğrenme, modelin girdiler \mathbf{S} ve çıktılar \mathbf{Y} arasındaki eşleşme ilişkisi $f : \mathbf{X} \rightarrow \mathbf{Y}$ ’yi öğrenmesi amacıyla önceden etiketlenmiş veri setleri kullanılarak eğitildiği bir yöntemdir. Model, bu ilişkiyi öğrenerek, yeni ve görülmemiş veriler üzerinde doğru tahminlerde bulunmayı amaçlar. Sınıflandırma, belirli özelliklere sahip verilerin önceden tanımlanmış sınıflara ($y_i \in \{1, 2, \dots, K\}$) ayrılmasını ifade eder. Öğrenme, daha bilinen adı ile eğitim, gerçek sınıflar y ve tahmin edilen sınıflar \hat{y} arasındaki hatanın fonksiyonu olan bir hata (kayıp - loss) fonksiyonu $L(y, \hat{y})$ ’yi en küçükleyen f ilişkisinin belirlenmesi sürecidir.

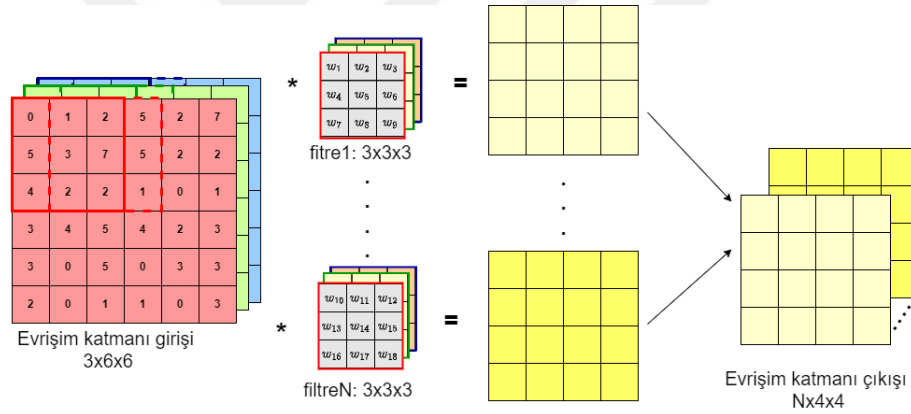


Şekil 2.4 Deney 18 için PWR doppler spektrogramları (a) GK2, (b) GK3, (c) GK4

Son yıllarda, daha karmaşık ve büyük veri setleriyle başa çıkmak için derin öğrenme teknikleri geliştirilmiştir. Derin öğrenme modelleri, katmanlı yapıları ile özellikle görsel ve metin tabanlı verilerdeki karmaşık örüntüleri tanıyarak yüksek doğruluk oranları ile sınıflandırma yapabilmektedir. Derin öğrenme modellerini, klasik makine öğrenmesi (Machine Learning - ML) yöntemlerinden ayıran temel özelliklerden biri öznelik (features) çıkarım yaklaşımıdır. ML yöntemlerinde, modelin iyi performans göstermesi için gerekli özneliklerin çoğunlukla insan müdahalesi (hand-crafted) ile tasarlanması ve seçilmesi gereklidir. Bunların, veri setine, kullanılacak yönteme ve uygulama alanına uygun olarak belirlenmesi performansı etkileyen en önemli etmenlerdendir. Derin öğrenme modelleri ise verilerden, ilgili uygulama için en uygun olacak öznelikleri öğrenerek kendi özellik çıkarım yapılarını oluşturabilirler. Bu, özellikle görsel veriler için büyük bir avantaj sağlar. Derin öğrenmede temel olarak üç katman bulunmaktadır: girdi katmanı, çıktı katmanı ve gizli (ara) katmanlar. Girdi katmanı, modele beslenen ham verilerin (tez kapsamında, bir görüntüdeki piksel değerleri) alındığı katmandır. Çıktı katmanı ise modelin tahmin etmek istediği sonuçları üreten katmandır. Sınıflandırma problemlerinde bu katman, her bir sınıf için (tez kapsamında 7 farklı aktivite sınıfı) bir nöron içerir ve her nöronun çıktısı, modelin ilgili sınıfa ait tahmin ettiği olasılığı temsil eder. Gizli katmanlar, giriş verileri ile çıktılar arasındaki ilişkiyi öğrenmek için kullanılan ara katmanlardır ve farklı yapılar ve işlevlerde olabilirler. Bu çalışmada özellikle görsel verileri işlemek için tasarlanmış özel bir tür derin öğrenme modeli olan Evrişimli sinir ağları (Convolutional Neural Network-CNN) kullanılmıştır. CNN yapıları evrişim katmanı (convolutional layer),

havuzlama katmanı (pooling layer) ve tam bağlantılı katmanı (fully connected layer) gibi belirli katman türlerini kullanarak veri içerisindeki örüntüleri çıkarır. CNN'leri oluşturan temel katmanlar ve bileşenler şu şekilde sıralanabilir:

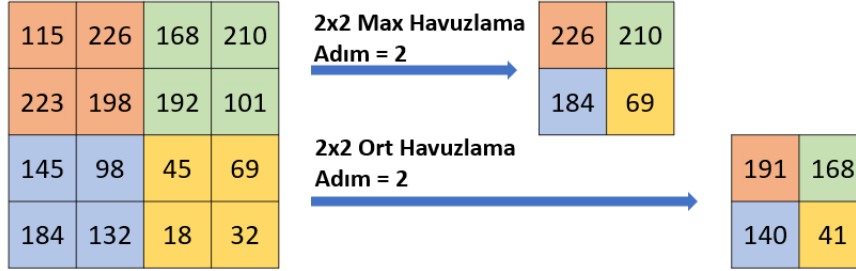
- **Evrişim Katmanı:** CNN'in en temel bileşeni olan evrişim katmanı, girdi verisine çeşitli boyut ve sayıda filtreler uygulayarak veriden öznitelikler çıkarmak için kullanılır. Şekil 2.5, bir evrişim katmanının çalışma yapısını 3 kanallı 6x6'lık bir giriş için göstermektedir. Burada N adet 3x3x3 boyutunda filtre kullanılmıştır. Filtre kanal sayısının giriş kanal sayısı ile aynı olması gerekmektedir. Filtreler giriş üzerinde kaydırılırken kaydırma adım sayısı (stride) 1 olarak seçilmiştir ve her bir filtre için konvolüsyon işlemi 4x4 boyutunda bir çıktı üretmektedir. Bu çıktılar birleşerek evrişim katmanının çıkışını (Nx4x4) meydana getirir. Görüldüğü gibi filtre sayısı, çıkış kanal sayısını belirlemektedir. Birden fazla filtre kullanılması ile veri içerisindeki farklı özellikleri ve örüntüleri yakalanması sağlanmaktadır.



Şekil 2.5 N filtreden oluşan bir evrişim katmanının genel işleyiş adımları

- **Aktivasyon Fonksiyonu:** Genellikle her evrişim işleminden sonra uygulanan aktivasyon fonksiyonu, katmanın çıktılarını sınırlandırarak ve lineer olmayan dönüşümler uygulayarak, modelin doğrusal olmayan özellikler yakalamasını sağlar. En yaygın kullanılan aktivasyon fonksiyonu ReLU (Rectified Linear Unit)'dur. ReLU, negatif girdileri sıfıra eşitlerken, pozitif girdileri doğrudan çıktı olarak verir. Bu, eğitim sürecini hızlandırır, gradyan kaybolmasını önler ve ağıın karmaşık örüntü ve ilişkileri öğrenmesini sağlar.
- **Havuzlama (Pooling) Katmanı:** Genellikle evrişim ve aktivasyon katmanlarından sonra, hesaplama yükünü azaltmak ve aşırı uyumu (overfitting) önlemek amacıyla, girdi boyutunu küçültmek için kullanılır. Maksimum (Max) ve Ortalama Havuzlama en sık kullanılan havuzlama

yöntemlerindedir. Max Havuzlama, bir özellik haritasındaki belirli bir pencere boyutu (filtre boyutu) içerisindeki maksimum değeri alır. Bu, modelin küçük deformasyonlara ve yer değiştirmelere karşı daha dayanıklı olmasını sağlar. Ortalama Havuzlama ise, belirlenen pencere boyutu içerisindeki tüm değerlerin ortalamasını alır ve çıkışa tek bir değer döndürür. Şekil 2.6'de adım sayısı 2, pencere boyutu 2x2 olan bir max havuzlama ve ortalama havuzlama katmanı örneği gösterilmektedir.

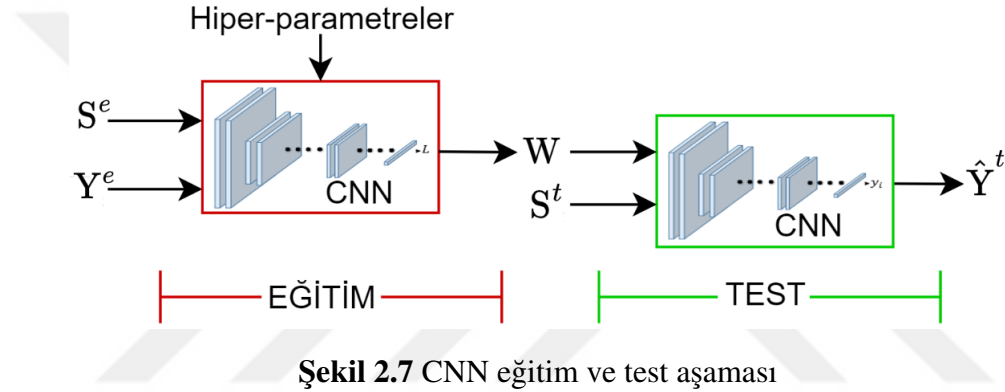


Şekil 2.6 4x4'lük bir görüntü için max ve ortalama havuzlama işlemleri

- **Tam Bağlantı Katmanı:** Evrişim veya havuzlama katmanlarının çıktıları tek boyutlu bir vektöre dönüştürülerek (flatten), tam bağlantı katmanına giriş olarak verilir. Bu katmanlar, evrişimli katmanların aksine, her girdi birimini her çıktı birimiyle bağlar ve girdi sınıflandırma gibi belirli bir çıktıyı üretmek için kullanılır.
- **Softmax katmanı:** Çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılır. Bu katman, önceki katmada üretilen değerleri girdi olarak alır ve her sınıfın tahmini olasılığını, bu olasılıkların toplamı bir olacak şekilde belirler. Böylece, modelin en olası sınıf tahmini son karar olarak belirlenmiş olur.

Bir CNN mimarisinin temel katman ve fonksiyonlarından bahsettikten sonra Şekil 2.7'de, modelinin eğitim ve test süreçlerini gösteren akış diyagramı sunulmaktadır. Görüldüğü üzere eğitim fazı, modelin eğitim verileri ve bunlara ait gerçek etiketleri içeren (S^e, Y^e) seti girdi olarak alır. Eğitim sırasında, gerçek çıktılar Y^e ile model tahminleri \hat{Y}^e arasındaki fark, modelin hata oranını, hata fonksiyonu aracılığı ile belirler. Bu aşamada, modelin performansını etkileyen hiper-parametreler, eğitim sürecinin verimliliğini ve etkinliğini optimize etmek amacıyla ayarlanır. Eğitim ileri besleme (feedforward) ve geri yayılım (backpropagation) süreçlerinden oluşmaktadır. İleri besleme, giriş verilerinin ağın giriş katmanından, gizli katmanlara ve en sonunda da çıkış katmanına doğru ağ boyunca ilerlediği süreci ifade eder. Geri yayılım ise ileri besleme ile elde edilen tahmin ve gerçek değerler arasındaki hatanın

hesaplanması ve çıkıştan girişe doğru her katmandaki ağırlıkların, hatayı azaltacak şekilde güncellenmesi sürecini ifade etmektedir. Burada, her bir ağırlığın, hata oranını nasıl etkilediğinin belirlenmesi ve ağırlıkların, hatanın azaldığı yönde adım adım güncellenmesi gerekmektedir. Burada güncelleme için farklı optimizasyon algoritmaları kullanılmaktadır. Modelin eğitimi tamamlandığında, veriyi en iyi modelleyen W ağırlıkları öğrenilmiş olur. Eğitim süreci, hata fonksiyonu ve optimizasyon algoritması Bölüm 2.3.4’de ayrıntılandırılmaktadır. Öğrenilen bu ağırlıkların sayısı, kullanılan derin öğrenme modeline göre farklılık göstermekle birlikte, tez kapsamında değerlendirilen üç modele ait bilgiler sonraki bölümlerde ayrıntılı açıklanmaktadır. Test aşamasında, model daha önce görülmemiş test verilerini (S^t) girdi olarak alır ve çıktılar (\hat{Y}^t) ileri besleme ile belirlenir. Performans, gerçek çıktılar (Y^t) ile modelin tahminleri kullanılarak hesaplanır.



Şekil 2.7 CNN eğitim ve test aşaması

Çalışmada, aktivite sınıflandırma için farklı sayıda katman içeren derin öğrenme mimarileri kullanılarak eğitim ve testler yapılmış ve sınıflandırma performansları karşılaştırılmıştır. Kullanılan AlexNet [29] ve VGG16 [30] mimarileri görsel sınıflandırma problemleri için evrişimli sinir ağlarını kullanan ilk yöntemlerdendir. 2012 yılında önerilen AlexNet, 5 evrişimli ve 3 tam bağlı katmanı olmak üzere 8 katmandan oluşmaktadır. Diğer yandan, AlexNet’in geliştirilmiş versiyonu olarak kabul edilen VGG16, 13 evrişimli ve 3 tam bağlı katmandan oluşur. Bunlardan farklı olarak ResNet, *kaybolan gradyan* sorununu ortadan kaldırarak daha derin evrişimli sinir ağlarının eğitilmesine olanak sağlamaktadır [31]. Bu çalışmada 18 ve 34 katmanlı iki farklı ResNet mimarisi kullanılmaktadır. Sonraki bölümlerde üç yöntemin, mimarileri, eğitim ve test süreçleri ayrıntılandırılmaktadır.

2.3.1 AlexNet mimarisi ile Sınıflandırma

AlexNet, 2012 yılında geliştirilen derin öğrenme ve bilgisayarlı görüde dönüm noktası olan bir sinir ağı mimarisidir. 5 evrişim katmanı ve 3 tam bağlı katman olmak üzere toplamda 8 katmandan oluşur. AlexNet ile tasarlanmış yaklaşık 57 milyon parametreye sahip sınıflandırıcı ağ mimarisi Tablo 2.3’te gösterilmektedir.

Görüldüğü gibi, farklı katmanlarda 11x11, 5x5, ve 3x3 boyutlarında çeşitli evrişim filtreleri kullanır. Bu, modelin farklı ölçeklerdeki özellikleri algılamasına olanak tanımaktadır.

Bu çalışmada, AlexNet, çok sınıflı sınıflandırma problemleri için yaygın olarak kullanılan çapraz entropi kayıp fonksiyonunu (cross-entropy loss function) kullanılarak eğitilmiştir. Bu fonksiyon, modelin çıktılarının gerçek etiketlere olan benzerliğini ölçer ve modelin doğru tahminler yapabilmesi için gereken ayarlamaları yapar. Çapraz entropi, modelin ürettiği olasılık dağılımları ile gerçek dağılım arasındaki farkı minimize eder, bu sayede modelin sınıflandırma performansı artırılır.

Tablo 2.3 AlexNet sınıflandırıcı mimari ve parametreleri

Katman adı	Çıkış boyutu	Filtre boyutu	#parametre
Conv + ReLU	64 x 49 x 49	11 x 11	23,296
Max Havuzlama	64 x 24 x 24	3 x 3	-
Conv + ReLU	192 x 24 x 24	5 x 5	307,392
Max Havuzlama	192 x 11 x 11	3 x 3	-
Conv + ReLU	384 x 11 x 11	3 x 3	663,936
2 x (Conv + ReLU)	256 x 11 x 11	3 x 3	1,475,072
Max Havuzlama	256 x 5 x 5	3 x 3	-
Ortalama Havuzlama	256 x 6 x 6	-	-
TamB. + ReLU	1 x 4096	-	37,752,832
TamB. + ReLU	1 x 4096	-	16,781,312
Tam bağlantı + Softmax	1 x 7	-	28,679

2.3.2 VGG16 mimarisi ile Sınıflandırma

VGG16, 2014 yılında Oxford Üniversitesi'nin Visual Geometry Group tarafından tasarlanmıştır ve bilgisayarlı görü alanında derin öğrenme modelleri arasında önemli bir yere sahiptir. Model, 13 evrişim katmanı ve 3 tam bağlı katman olmak üzere toplam 16 katmandan oluşur. VGG16, özellikle 3x3 lük küçük evrişim filtreleri kullanarak, daha derin bir ağ yapısında yüksek seviyede özellik çıkarımı yapabilme kapasitesine sahiptir. Bu küçük filtre boyutu, ağın farklı katmanlarında daha fazla ağırlık ve öğrenilebilir parametre içermesine olanak tanır, bu da modelin daha detaylı ve incelikli özellikleri öğrenmesine yardımcı olur. Modelin mimarisine ait ayrıntılar Tablo 2.4'de gösterilmektedir.

VGG16 modeli, büyük ve çeşitli veri kümeleri üzerinde eğitildiğinde, özellikle görsel nesne tanıma görevlerinde etkileyici sonuçlar göstermiştir. Model, sınıflandırma problemlerini çözerken AlexNet'te olduğu gibi çapraz entropi kayıp fonksiyonu ile optimize edilir. Ancak, VGG16'nın kullanımı, daha derin ve

tekrarlanan yapıları sayesinde, görsel sınıflandırma görevlerinde AlexNet'e göre genellikle daha iyi performans sergiler. VGG16, aynı zamanda, daha fazla öğrenilebilir parametre içerir ve bu da modelin daha büyük ve çeşitli veri kümeleri üzerinde daha iyi öğrenmesine yardımcı olur. Bu özellikler, VGG16'yı özellikle karmaşık görsel sınıflandırma problemleri için daha uygun bir seçenek haline getirir.

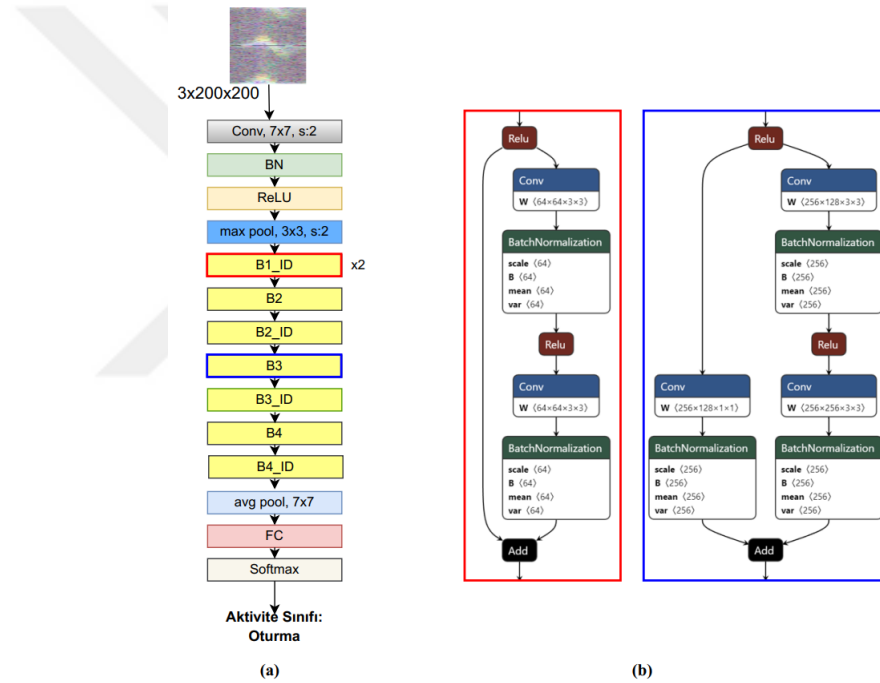
Tablo 2.4 VGG16 sınıflandırıcı mimari ve parametreleri

Katman adı	Çıkış boyutu	Filtre boyutu	# parametre
2 x (Conv + ReLU)	64 x 200 x 200	3 x 3	38,720
Max Havuzlama	64 x 100 x 100	3 x 3	-
2 x (Conv + ReLU)	128 x 100 x 100	3 x 3	221,440
Max Havuzlama	128 x 50 x 50	3 x 3	-
3 x (Conv + ReLU)	256 x 50 x 50	3 x 3	1,475,328
Max Havuzlama	256 x 25 x 25	3 x 3	-
3 x (Conv + ReLU)	512 x 25 x 25	3 x 3	5,899,776
Max Havuzlama	512 x 12 x 12	3 x 3	-
3 x (Conv + ReLU)	512 x 12 x 12	3 x 3	7,079,424
Max Havuzlama	512 x 6 x 6	3 x 3	-
Ortalama Havuzlama	512 x 7 x 7	-	-
TamB. + ReLU	1 x 4096	-	102,764,544
TamB. + ReLU	1 x 4096	-	16,781,312
Tam bağlantı + Softmax	1 x 7	-	28,679

2.3.3 Artık Ağ Mimarisi ile Sınıflandırma

Artık Ağ (Residual Network-ResNet) farklı katman sayılarından oluşan, sınıflandırma, nesne sezme, nesne takibi gibi bir çok alanda giriş özniteliklerinin çıkarımı amacıyla sıklıkla kullanılan bir derin öğrenme mimarisini ifade eder. VGG16 modelinin önerilmesi ile ortaya çıkan katman sayısı ile performans arasında her zaman pozitif bir ilişki olduğu görüşü, derinlik arttıkça doğruluk oranı düşmesi ile çürütülmüş oldu. Bu bozulmanın ana nedeni olan *kaybolan gradyan* sorunu modelin parametrelerini güncellemek için geri yayılım sırasında kullandığımız gradyanların hızla sifira yaklaşması ve özellikle ilk katmandaki parametrelerin bir noktadan sonra güncellenememesidir. ResNet, 2015 yılında bu probleme çözüm olarak önerilmiştir. Bu amaçla, iki evrişim katmanından sonra uygulanan kısayol diğer adı ile artık bağlantı (shortcut, skip veya residual connection) ile önceki evrişim katmanının çıktısı olan özellik haritası, ilerideki katmana eklenir. Kısayol bağlantısının kullanıldığı mimariler, içerdikleri katman sayısına göre ResNet18, ResNet34, ResNet50, ResNet101, ResNet152 olarak adlandırılmaktadırlar. Bu bölümde ResNet genel yapısı ResNet18 üzerinden ayrıntılandırılmıştır.

ResNet mimarilerinde her bir temel blok "bottleneck (BON)" olarak adlandırılır ve her bir temel blok 2 ya da 3 evrişim katmanı (ResNet18 ve ResNet34 için 2) içerir. Bu blokların çıktısı, blok girişi ve son evrişim katmanınının toplanması ile elde edilmektedir. Başka bir deyişle, kısayol bağlantısı kullanmayan mimarilerde blok çıkışı, blok içerisinde gerçekleştirilen fonksiyon F ve giriş özneliği x , olmak üzere $H(x) = F(x)$ iken, artık ağlarda, kısayol bağlantı fonksiyonu S olmak üzere, $H(x) = F(x) + S(x)$ olur. Bu gradyanların ilk katmanlara doğru çok küçülmesini engelleyerek, gradyanların kaybolması problemini büyük oranda çözer. Temel bloklarda, evrişim katmanlarından sonra,"batch" normalizasyonu (BN) [36]ve ReLU aktivasyon fonksiyonu kullanılır. BN, eğitim sırasında batch içindeki verilerin çıktılarının normalize edildiği katmandır. Bu şekilde eğitim süreci hızlandırılmakta ve model performansı artırılmaktadır.



Şekil 2.8 (a) ResNet18 mimarisi (b) Soldan sağa, Birinci ana temel blok, birim bağlantılı BON (B1-ID), Üçüncü ana temel blok, izdüşüm bağlantılı BON (B3)

Şekil 2.8’da toplam 18 katmandan oluşan, 5 ana katman bloğu içeren ResNet18’in genel mimarisi, konvolüsyon katmanları ve kısayol bağlantısı içeren iki farklı BON için ayrıntılar verilmiştir. ResNet18’in ilk katman bloğunda (Şekil 2.8(a)- ilk dört katman) herhangi bir atlama bağlantısı bulunmaz. Diğer dört ana katman bloklarında ise 2’şer adet, her biri bir atlama bağlantısı bulunduran BON bloğu yer almaktadır. Son katman çıkışından sonra bir ortalama havuzlama katmanı ve tam bağlantı katmanı (FC) ile elde edilen 512 kanallı bir öznelik haritası (Tablo 2.5), SoftMax katmanına iletilerek, mevcut sınıflara ait olma olasılıkları hesaplanır (Bu tez çalışması kapsamında 7 sınıf için 7 değer).

Tablo 2.5 ResNet18 sınıflandırıcı mimari ve parametreleri

Katman/Blok adı	Çıkış boyutu	Filtre boyutu	#parametre
Conv	64 x 100 x 100	7 x 7	9,408
BatchNorm + ReLU	64 x 100 x 100	-	128
Max Havuzlama	64 x 50 x 50	3 x 3	-
2 x B1 - ID	64 x 50 x 50	3 x 3	147,968
B2	128 x 25 x 25	3 x 3	230,144
B2 - ID	128 x 25 x 25	3 x 3	295,424
B3	256 x 13 x 13	3 x 3	919,040
B3 - ID	256 x 13 x 13	3 x 3	1,180,672
B4	512 x 7 x 7	3 x 3	3,673,088
B4 - ID	512 x 7 x 7	3 x 3	4,720,640
Ortalama Havuzlama	512 x 1 x 1	7 x 7	-
TamB. + Softmax	1 x 7	-	3,591

ResNet mimarisinde iki artık bağlantı türü vardır. Şekil 2.8(b)-ilk görselde görüldüğü gibi blok girişini direkt olarak blok çıkışına aktaran artık bağlantı birim bağlantı (identity connection - ID block) olarak adlandırılır. Burada $S(x) = x$ olur ve blok çıkışı $H(x) = F(x) + x$ olarak hesaplanır. Bu bağlantılar, blok girişi (x) ve çıkışının ($H(x)$), aynı boyutlara sahip olması durumunda kullanılır. Blok çıkışındaki kanal sayısının, blok girişindeki kanal sayısına eşit olmadığı bloklarda ise toplama işlemini gerçekleştirebilmek amacıyla, Şekil 2.8(b)-ikinci görselde görüldüğü gibi kısayol bağlantısında 1×1 lik bir evrişim katmanını kullanılarak girişten gelen verinin kanal sayısı, katmanların çıkışındaki verinin kanal sayısına eşitlenir. Burada S , kanal eşitlemesini yapan izdüşüm fonksiyonu olmak üzere, blok çıkışı $H(x) = F(x) + S(x)$ olarak hesaplanır ve izdüşüm bağlantısı (projection connection) olarak adlandırılır. Şekil 2.8(b)'de, ResNet18'in Temel Blok 1 (B1-ID) ve Temel Blok 3 (B3) 'e ait BON'ların bağlantıları gösterilmiştir. Tüm izdüşüm bağlantıları 1×1 'lik evrişim katmanlarından oluşmak üzere, temel blok filtre boyutları her blok için farklılık gösterebilir.

Bu tez çalışmasında ResNet18 ve ResNet34 mimarileri kullanılmış ve performansları raporlanmıştır. ResNet18'in genel mimarisi ve bloklardaki filtre boyutları Tablo 2.5'de parametere sayıları ile birlikte gösterilmiştir. Görüldüğü gibi tüm birim blokların (ID) girişleri, çıkış boyutu ile aynı iken, diğer bloklarda farklıdır. Parametreler, diğer mimarilerde olduğu gibi giriş boyutu $3 \times 200 \times 200$ 'lük görüntü için hesaplanmıştır. ResNet18 ile benzer olarak 5 ana temel blok içeren ResNet34 toplam 34 temel katmandan oluşur. Benzer şekilde ilk katman bloğunda herhangi bir atlama bağlantısı bulunmaz. Farklı olarak, 1., 2., 3. ve 4. katman blokları sırasıyla 3, 4, 6 ve 3 adet, BON bloğundan oluşur.

2.3.4 Derin Öğrenme Mimarilerinde Kayıp Fonksiyonu ve Eğitim Süreci

Derin öğrenme mimarilerinde eğitim, bir modelin x girdileri (tez çalışmasında spektrogramlar) ve y çıktıları (tez çalışmasında aktivite sınıfları) arasındaki ilişkiyi öğrenmesini sağlamak amacıyla ağırlıkların optimize edilmesidir. Bu süreçte kayıp fonksiyonu (loss function), x girişi için tahmin edilen çıktı \hat{y} ve beklenen çıktı y arasındaki hatanın bir fonksiyonudur. Amaç kayıp fonksiyonunun, dolayısı ile hatayı en küçükleyen optimum model parametrelerinin belirlenmesidir. Kayıp fonksiyonu ve parametrelerinin belirlenme sürecinde kullanılacak optimizasyon yöntemi, modelin doğruluğunu ve genelleme yeteneğini doğrudan etkiler. Tez çalışmasında tüm derin öğrenme modelleri için, çok sınıflı sınıflandırma problemlerinde yaygın olarak kullanılan karşıt ilinti kayıp fonksiyonu (Cross Entropy Loss) tercih edilmiştir. Kayıp fonksiyonu, denklem 2.1'de verilmiştir.

$$L = - \sum_{c=1}^C y_c \log(\hat{p}_c) \quad (2.1)$$

Burada;

- C toplam sınıf sayısını (tez kapsamında $C = 7$),
- y_c , c . sınıfa ait gerçek değeri (Ground truth-GT, GT sınıf için 1, diğer sınıflar için 0),
- \hat{p}_c sınıflandırıcı çıkışında hesaplanan c . sınıfa ait olasılık değeridir.

Denklem 2.1'deki logaritma fonksiyonu sayesinde, GT sınıf için hesaplanan olasılık değeri arttıkça (istenen durum) kayıp fonksiyonunun değeri hızla düşer. Örneğin, model ile gerçek sınıfa ait tahmin edilen olasılık değeri $\hat{p}_c = 1$ olması durumunda $\log(\hat{p}_c) = 0$ ve dolayısı ile kayıp değeri 0 olarak hesaplanacaktır. Aksine $\hat{p}_c = 0.01$ gibi istenmeyen düşüklükte bir değer olduğunda $\log(\hat{p}_c) = -2$ ve kayıp değeri 2 olacaktır. Bu kayıp değeri, parametrelerinin belirli bir oranda güncellenmesi gerektiğini ifade etmektedir. Güncellenmenin nasıl yapılacağı optimizasyon algoritması ile belirlenir ve tez çalışması kapsamında, bu güncellemeler için, derin öğrenme modellerinin eğitiminde yaygın olarak kullanılan ADAM [37] optimizasyon yöntemi kullanılmıştır. Eğitim sürecinde, modelin çıktılarının olasılık dağılımına dönüştürülmesi için (\hat{p}_c) Softmax aktivasyon fonksiyonu kullanılır. Softmax fonksiyonu, modelin FC çıktısında her bir sınıf için ürettiği ham değerleri (z_c) normalleştirilmiş olasılıklara dönüştürür ve Denklem 2.2 ile hesaplanır.

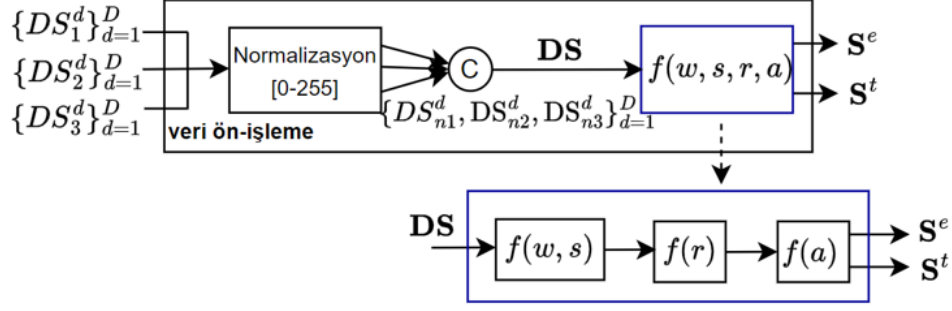
$$\hat{p}_c = \frac{e^{z_c}}{\sum_{j=1}^C e^{z_j}} \quad (2.2)$$

Derin öğrenme modellerinin eğitiminde, verilerin tamamını tek seferde veya her veriyi tek tek işlemek hesaplama açısından zorlayıcı olabileceği gibi modelin genelleme performansını da olumsuz etkiler. Bu nedenle, veriler genellikle, "batch" olarak adlandırılan daha küçük gruplara bölünerek işlenir. Her batch, belirli sayıda eğitim örneği içerir ve bu örnekler üzerinden kayıp fonksiyonu, batch içindeki örneklerin ortalaması alınarak hesaplanır. Batch kullanımı, eğitim sürecini hızlandırmak ve bellek kullanımını optimize etmek için yaygın bir tekniktir.

Bir derin öğrenme modelinin eğitim süreci kısaca şu adımları içerir: ilk adım, modelin başlangıç ağırlıklarının belirlenmesidir. Tez çalışması kapsamında ImageNet veri seti [38] ile eğitilmiş ön-ağırlıklar kullanılmıştır. Daha sonra, her iterasyonda bir batch, model tarafından işlenerek (ileri besleme) kayıp fonksiyonu ve kayıp fonksiyonunun modelin ağırlıklarına göre türevleri (gradyanlar) hesaplanır (geri yayılım süreci). Bu gradyanlar, modelin hata yüzeyinde hangi yönde ve ne kadar ilerlemesi gerektiğini gösterir. Optimizasyon algoritması (çalışmada ADAM) kullanılarak ağırlıklar, hesaplanan gradyanlar doğrultusunda iteratif olarak güncellenir. Ağırlık güncellemeleri iterasyon olarak isimlendirilmekte ve her batch için yapılmaktayken, tüm eğitim veri setinin model tarafından bir kez tamamen işlenmesi "epoch" olarak isimlendirilir. Başka bir deyişle, bir epoch, birden çok iterasyon içerir ve her iterasyonda bir batch işlenir. Belirlenen epoch sayısı tamamlandığında veya kayıp fonksiyonu belirli bir eşiğin altına düştüğünde eğitim süreci sonlandırılır. Tez kapsamında kullanılan batch ve epoch sayısı gibi parametreler, performans testleri bölümünde ayrıntılı olarak verilmiştir.

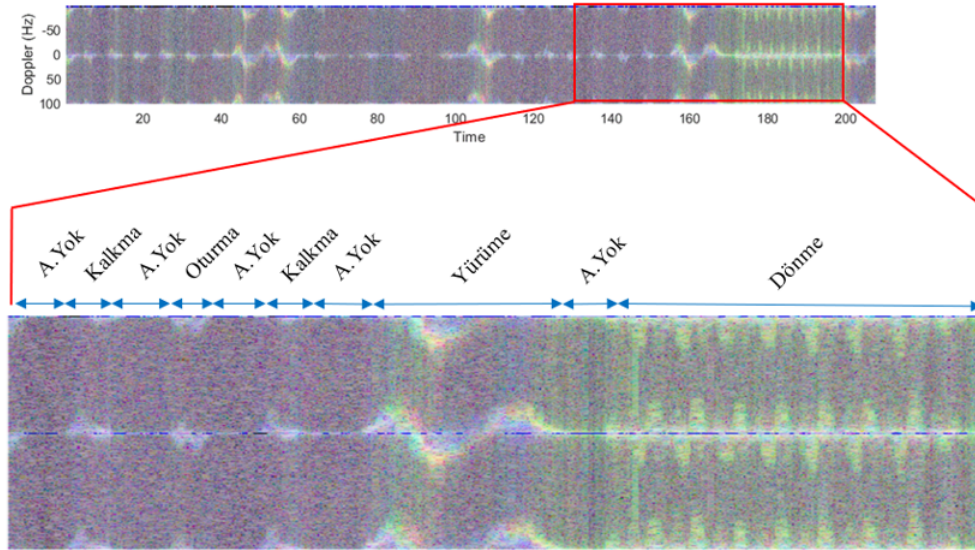
2.4 Veri Ön-işleme ve İyileştirme Yaklaşımı

Veri ön-işleme ve iyileştirme, aktivite tanıma performansını iyileştirmek amacıyla, ham veriyi makine öğrenimi modellerine uygun hale getirme ve verinin kalitesini ve sayısını artırma adımlarını içermektedir. Bu işlemler, modelin genelleme yeteneğini artırarak daha güvenilir ve doğru sonuçlar elde edilmesini sağlamaktadır. Bu çalışmada ham veri, her bir gözlemci kanaldan alınan verinin spektrogramları olarak tanımlanabilir. Şekil 2.9'da d . deney için DS_1^d , DS_2^d ve DS_3^d olarak ifade edilen doppler spektrogramlarından, eğitim ve test veri setlerinin oluşturulmasına kadar gerçekleştirilen ön işleme adımlarına ait akış görülmektedir. Bu bölümde şekilde $f(w, s)$, $f(r)$ ve $f(a)$ fonksiyonları ile ifade edilen teknikler ayrıntılandırılacaktır.



Şekil 2.9 Her bir deney için, spektrogramların eğitim ve test verisine dönüştürülmesi

Veri setinin oluşturulması amacıyla öncelikli olarak, bir önceki bölümde bahsedildiği gibi, her biri farklı sürelerde (L saniyelik) 36 deney için 3 gözlemci kanaldan 3 doppler spektrogramı DS_1^d, DS_2^d ve DS_3^d oluşturulmuştur. Burada d deney numarasını göstermektedir. Bu spektrogramlar, verilerin ölçeğindeki farklılıkları azaltmak ve öğrenme hızını arttırmak amacıyla, 0-255 arasına normalize edilmektedir ($DS_{ni}^d, i=1,2,3$). Bu işlemin ardından, spektrogramlar kanal bazında birleştirilmekte ve $DS^d = \text{concat}(DS_{n1}^d, DS_{n2}^d, DS_{n3}^d)$, d . deneye ait üç kanallı spektrogram görüntüsü olmak üzere, tüm deneylerin 3 boyutlu (3B) görüntülerini içeren $DS = (DS^1, \dots, DS^{36})$ elde edilmektedir. Bölüm 2.2.2’de her bir spektrogramı ayrı ayrı işlemek yerine, 3 boyutlu spektrogram ile aktivite tanıma yapılmasının gerekliliği gösterilmiştir (Şekil 2.4). Bu şekilde, modelin aynı anda daha fazla bilgiyi kullanması ve farklı kanallar arasındaki ilişkileri öğrenmesi sağlanarak performansın artırılması amaçlanmaktadır.



Şekil 2.10 Deney 18 için 3 kanallı pwr doppler spektrogramı ve farklı aktive sınıfları için spektrogram verisinin değişimi

Şekil 2.10, Şekil 2.4’de verilen Doppler spektrogramlarının 3 boyutlu olarak birleştirilmiş halini (DS^{18} ’in 207 saniyelik bir kesiti) ve farklı aktivite sınıfları için bu 3B spektrogramın nasıl değiştiğini göstermektedir. Anlaşılabilirliği artırmak amacıyla, aktivite değişimleri, 3B spektrogramın 70 saniyelik kısmında gösterilmiştir. Şekilde görüldüğü gibi, farklı aktivitelerin spektrogramlarının sahip olduğu ayırt edici özellikler PWR verilerinden elde edilen spektrogramların aktivite tanıma için kullanılabilir olduğunu net olarak göstermektedir. Bu noktadan sonra, formülasyonları sadeleştirilmek amacıyla, d deney numarası indisi kaldırılmış ve ifadeler tek bir deney için verilmiştir, ancak her deney için aynı işlemler uygulanmakta eğitim ve test veri seti tüm deneyleri içermektedir.

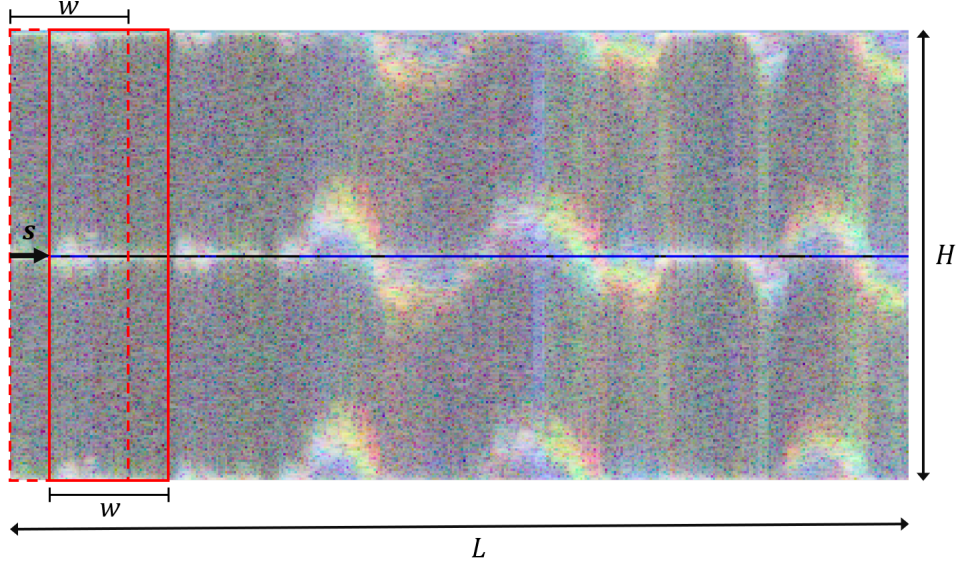
Her deney için elde edilen 3B spektrogramların, aktivite tanıma için kullanılabilmesi amacıyla sırasıyla;

- Gerçekleştirilen aktivitelerin anlık olarak belirlenmesi için, spektrogramların alt bölütlere ayrılması ($f(w, s)$)
- Bölütlerin aktivite etkilemelerinin belirlenmesi ($f(r)$)
- Oluşturulan bölütlerin eğitim ve test seti olarak ayrılması ($f(a)$),

adımları uygulanır.

Açıklandığı gibi, her bir deney farklı uzunluklarda ve çeşitli sayı ve sıralarda aktiviteler içermektedir. Gerçekleştirilen aktivitelerin belirlenmesi için, spektrogramların bölütlere ayrılması gerekmektedir. Bu işlemin nasıl gerçekleştirileceği, veri kalitesi ve veri sayısını önemli ölçüde etkilemektedir. Çalışmada bu amaçla $f(w, s)$ fonksiyonu önerilmiştir. Burada, belirli bir süre aralığında (w saniyelik) pencereler kullanılarak spektrogramlar bölütlenmektedir. Pencereleme işlemi sırasında $s \leq w$ olmak üzere s saniye kaydırma uygulanarak bölütler oluşturulmaktadır (Şekil 2.11). Bu noktada, s değerinin w değerinden küçük olması, pencerelerin örtüşmesini sağlarken, eşit olması her pencerenin yeni bir bölgeyi kapsadığını ve bir ölçümün yalnızca bir kez kullanılacağı anlamına gelir. Bu kaydırma parametresinin doğru olarak seçilmesi, bilgi kaybını en aza indirilmesi ve veri çeşitliliğinin artırılması açısından önemlidir. Bu sayede, sınırlı sayıda eğitim örneğine sahip olunan durumlarda veri miktarı artırılarak modelin kararlılığı ve doğruluğu artırılmaktadır.

Spektrogramdan elde edilmiş w saniyelik t . görüntü S_t olmak üzere, L saniyelik bir deney için bölünmüş görüntü kümesi \mathbf{S} denklem 2.3 ile ifade edilebilir.



Şekil 2.11 3B spektogramın w ve r parametrelerine göre bölütlenmesi ($f(w, r)$)

$$\mathbf{S} = \{S_t \mid t = 0, 1, 2, \dots, \left\lfloor \frac{L - w}{s} \right\rfloor\} \quad (2.3)$$

Burada, t . görüntü için pencere başlangıç konumu $\text{bas}_t = t \times s$ ve bitiş konumu $\text{son}_t = \text{bas}_t + w$ olmak üzere $S_t = \mathbf{DS}_{\text{bas}_t:\text{son}_t}$ olarak ifade edilebilir. Her bir pencere belirli bir süreyi temsil ettiğinden, w parametresi her bir aktivite tanıma işlemi için kaç saniyelik verinin kullanılacağını gösterir. Büyük w değerleri, modelin daha fazla veri kullanarak daha kararlı tahminler yapmasını sağlayabilmekle birlikte, bu durum kısa süreli aktivitelerin tespit edilmesini zorlaştırabilmektedir. s parametresi ise, gerçek dünyada aktivitelerin ne kadar sıklıkla tanınacağını ifade etmektedir. Küçük s değerleri, daha sık aktivite tanıma yapmayı sağlamaktadır. Bu, özellikle hızlı değişen aktivitelerin tespit edilmesi için önemlidir. İlgili parametre değerleri performans testleri bölümünde ayrıntılı olarak verilmiş ve bu değerlerin performansa etkileri ilgili bölümde tartışılmıştır.

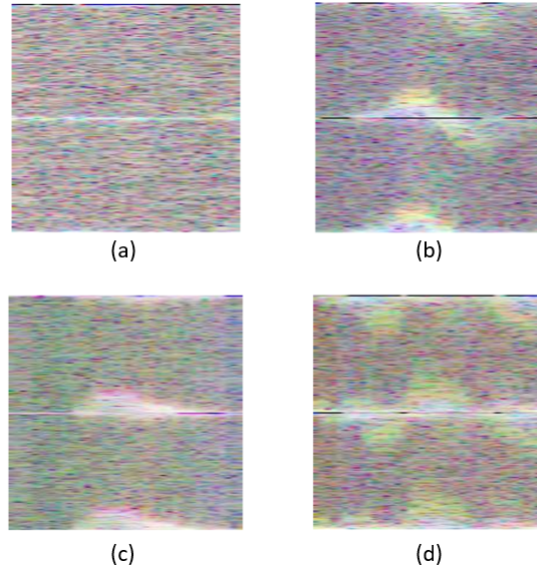
Elde edilen w saniyelik S_t spektogram bölütleri, 10 Hz'lik bir sistemde $10 \times w$ adet ölçüm içermektedir. Veri setinde her ölçüm için bir aktivite etiketi atanmıştır. Bu nedenle bölütler, doğrudan tek bir aktivite sınıfına atanamamakta, belli durumlarda, özellikle aktivite geçişlerinde, 2 veya daha fazla sınıf için bilgi içerebilmektedir. Bu çoklu sınıf bilgisi içeren durumlar, veri etiketleme sürecini karmaşıktır. Bir CNN yapısında eğitim verilerinin doğru bir şekilde etiketlenmiş olması, modelin performansını büyük ölçüde etkiler. Bu çalışmada, veri etiketleme sürecinin doğruluğunu artırmak için yeni bir $f(r)$ fonksiyonu ile etiketleme yapılması önerilmiştir. Bu yaklaşım ile, belirsizliğe sahip veri sayısı azaltılırken, etiketlenen verilerin güvenilirliği artırılmıştır. Her bir S_t bölütü için, bölütte en sık rastlanan

etiket p_{S_t} ve bu etiketin var olma süresi n_{S_t} (w sn lik bölütler için en çok w değerini alabilir) olmak üzere, ilgili bölüt için atanan etiket y_{S_t} denklem 2.4'de gösterilmektedir.

$$y_{S_t} = \begin{cases} p_{S_t} & \text{eğer } n_{S_t} > r \\ -1 & \text{aksi halde} \end{cases} \quad (2.4)$$

Burada, n_{S_t} , ön tanımlı bir güven eşik değeri r 'den büyükse, bu bölüt için gerçek etiket $y_{S_t} = p_{S_t}$ olarak belirlenir. Başka bir deyişle, bölütte en sık rastalanan sınıf, w saniyelik bölütün en az r saniyesinde yer alıyorsa o sınıf ile etiketlenir. Ters durumda ise, belirsiz olarak kabul edilir ve eğitim/test veri setinden çıkartılır. Sonuç olarak veri kümesi sadece etiketlenmiş veri ve sınıflarını (S_t, y_{S_t}) içerecek şekilde düzenlenmiş olur. r parametresinin doğru seçimi, modelin performansını doğrudan etkiler. Yüksek r değeri, yalnızca güvenilir ve net bir şekilde tanımlanmış bölütlerin etiketlenmesini sağlar, böylece modelin eğitiminde kullanılan verilerin kalitesi artırılır. Ancak, bu durum kullanılabilir veri sayısını azaltarak, model eğitimi için yeterli veri bulunmasını zorlaştırır. Ayrıca, özellikle aktivite geçişlerini içeren bölütlerin elenmesi, modelin geçiş anlarındaki ana aktiviteyi yakalayamamasına neden olur ve bu da gerçek dünya senaryolarındaki performansını düşürür. Öte yandan, daha düşük bir r değeri, daha fazla verinin etiketlenmesine olanak tanıyarak veri setinin genişlemesine katkıda bulunur. Ancak bu, etiketlemenin genel güvenilirliğinin azalmasına neden olabilmektedir. Derin öğrenme modellerinde giriş boyutunun sabit olması gerektiğinden, etiketlenen görüntüler, 3x200x200'lük boyuta getirilir. Bu amaçla, yeni piksellerin değerlerini en yakın dört pikselin ağırlıklı ortalaması olarak hesaplayan ve görüntü yeniden boyutlandırma sıklıkla tercih edilen bilineer enterpolasyon yöntemi kullanılmıştır. Şekil 2.12 belirli bir r değeri için farklı aktivite sınıflarına atanmış ve yeniden boyutlandırılmış spektrogram bölütlerini göstermektedir.

36 deney için etiketlenmiş veri kümesinin elde edilmesinin ardından, eğitim ($\mathbf{S}^e, \mathbf{Y}^e$) ve test ($\mathbf{S}^t, \mathbf{Y}^t$) setleri oluşturulur. Burada \mathbf{Y}^t ve \mathbf{Y}^e ilgili verilerin gerçek etiketleri y_{S_t} 'leri içeren kümeleri ifade etmektedir. Bu çalışmada $f(a)$ fonksiyonu, veri setini %a oranında eğitim ve kalan kısmını test setine rastgele olarak ayırmaktadır. Burada her sınıfın veri setindeki oranı korunarak sınıf dengesinin sağlanmasını hedeflenmiştir. Başka bir deyişle, her sınıf için eğitim/test oranı aynı kalacak şekilde ayırım gerçekleştirilmiştir. Böylece, eğitim ve test setlerinin her ikisi de veri setinin genelini temsil eder ve her sınıf için yeterli sayıda örnek içerir. Sınıf dengesinin korunması, özellikle dengesiz veri setlerinde modelin tüm sınıfları öğrenme ve genelleme yeteneğini artırır. Farklı parametreler için oluşturulan eğitim ve test seti boyutları performans testleri bölümünde ayrıntılı olarak raporlanmıştır.



Şekil 2.12 Farklı sınıflar için spektrogram bölütleri (a) Aktivite Yok, (b) Yere Uzanma, (c) Ayakta Durma (d) Dönme

2.5 Aktivite sınıflandırma için uçta uygulama

Bu çalışmada, model eğitimi ve performans değerlendirmesi, yüksek hesaplama kapasiteli Google Colab NVIDIA T4 GPU'da gerçekleştirilmiştir. Bununla birlikte, modelin gerçek dünya koşullarındaki uygulanabilirliğini görmek amacıyla, test süreci uçta uygulama olarak, Raspberry Pi mini bilgisayarı üzerinde de yürütülmüş ve hız karşılaştırmaları farklı mimariler için yapılmıştır. Eğitimin sadece yüksek hesaplama kapasiteli bir bilgisayarda yapılmasının nedeni, eğitim sürecinin yüksek işlem gücü ve büyük veri setleri gerektirmesidir.

Raspberry Pi, düşük maliyetli, enerji verimli ve taşınabilir bir bilgisayar olması nedeniyle uçta uygulama için ideal bir donanım olarak sıklıkla tercih edilmektedir. Tablo 2.6'de kullanılan donanıma ait özellikler ayrıntılı olarak görülmektedir.

Uçta uygulamanın çalışmaları, özellikle aktivite tanıma gibi gerçek zamanlı veri işleme gerektiren uygulamalarda gecikme sürelerini minimize ederek büyük avantajlar sağlamaktadır. Bu çalışmada, eğitim sonrası elde edilen model ve spektrogramları oluşturan ölçümler, test süreci için Raspberry Pi'ye aktarılmıştır. Önerilen ön işleme ve iyileştirme adımları sonrasında test gerçekleştirilmektedir. Şu an için, PWR verileri sensörler aracılığı ile alınamamakta birlikte, gelecekte doğrudan aktarılmasıyla daha entegre ve gerçek zamanlı çözümler geliştirme potansiyelinin olduğu açıktır. Raspberry Pi üzerinde yapılan testler, modelin sınırlı donanım kaynaklarına rağmen kullanılabilirliğini göstermiştir. Derin öğrenme modellerinin hız ve aktivite tanıma performanslarının karşılaştırmaları *Performans Testleri* bölümünde ayrıntılı olarak verilmektedir.

Tablo 2.6 Raspberry Pi 5 donanım özellikleri

İşlemci	Broadcom BCM2712
İşlemci Hızı	2.4GHz
Çekirdek Tipi/Sayısı	Cortex-A76 64bit Armv8 / 4 Çekirdek
Grafik İşlemci(GPU)	VideoCore VII
Grafik İşlemci Hızı	800MHz
RAM	4GB LPDDR4X-4627
Hafıza	SDR104 High Speed-microSD



3

PERFORMANS TESTLERİ

3.1 Hiper-parametreler ve Performans Metrikleri

Gerçekleştirilen testlerde OperaNet veri setinde yer alan PWR sensörü verileri [26] kullanılmıştır. Veri seti, *yürüme*, *oturma*, *ayakta durma*, *yere uzanma*, *yerden kalma*, *dönme* olarak adlandırılan 6 aktivite sınıfına ait ölçümleri içermekte olup, ayrıca aktivite olmaması durumunda (*Aktivite yok*) alınan ölçümleri de içermektedir.

Eğitim ve test setleri Bölüm 2’de ayrıntılı olarak açıklandığı şekilde oluşturulmuştur. Burada, 6 aktivite sınıfı için kullanılan parametreler $w = 4$ sn, $s = 1$ sn ve $r = 2$ sn olarak belirlenmiştir. Dolayısı ile derin öğrenme mimarilerinde kullanılmak üzere oluşturulan veriseti 4 saniyelik spektrogramlardan oluşmaktadır. Bu sürenin, aktivite sınıflarının karakteristik özelliklerini yakalamak için yeterli bilgi sağladığı bilinmekle birlikte [32], bu durum yapılan testlerde doğrulanmıştır. Verilerin 10 Hz ölçüm hızında kaydedilmesi sebebi ile her saniye için 10 ölçüm değeri raporlanmıştır. Sonuç olarak 4 saniyelik spektrogram görüntüsü 40 adet ölçüm verisi içermektedir. s parametresinin 1 saniye olarak belirlenmesi, aktivite tanımının her 1 saniyede bir geçmiş 4 saniyelik verinin kullanılması ile gerçekleştirildiği anlamına gelmektedir.

Veri seti, aktivitenin olmama durumunda gerçekleştirilen ölçümleride içermektedir. Bunun yanı sıra, aktivitelerin başlangıç, bitiş ve aktiviteler arası geçiş anları da aktivite yok olarak etiketlenmiştir. Bunun sonucu olarak, aktivite yok sınıfına ait örneklerin sayısının çokluğu, veri setinde dengesizliğe yol açmakta ve performansı olumsuz etkilemektedir. Bu probleme çözüm olarak, aktivite yok sınıfında yalnızca yüksek güvenilirliğe sahip örneklerin veri setine dahil edilmesi önerilmiştir. Bu amaçla, Aktivite Yok sınıfı için güven eşiği $r_{ay} = 3.9$ saniye olarak seçilmiştir. Bu sayede, sınıflar arası dengenin sağlanması amaçlanmış ve derin öğrenme modelinin performansı artırılmıştır. Sonuç olarak bu yaklaşım, modelin hem eğitim hem de validasyon aşamalarında daha tutarlı ve güvenilir sonuçlar üretmesine olanak tanır.

Belirtilen parametreler ile oluşturulan veri seti toplam 11,235 adet spektrogram bölütünden oluşmaktadır. Ön işleme adımında da ayrıntılandırıldığı gibi eğitim/test ayrımı sırasında kullanılan $f(a)$ fonksiyonunda $a = 80$ yapılarak, veri setinin %80'i eğitim seti ve kalan %20'lik kısım test seti olarak rastgele ayrılmıştır. Burada vurgulanması gereken önemli nokta, ayrımın veri setinin istatistiksel özelliklerini ve sınıf dağılımlarını koruyacak şekilde yapılmış olduğudur. Bu amaçla, %80 - %20 bölme işlemi her bir sınıf için ayrı ayrı yapılmış, sonrasında eğitim ve test seti oluşturulmuştur.

Tez kapsamında, ön işleme ve iyileştirme adımları dışında derin öğrenme modellerinde kullanılan hiper-parametreler Tablo 3.1'te gösterilmektedir.

Tablo 3.1 Derin öğrenme modelleri hiper-parametreleri

Parametre	Değeri
Epoch sayısı	100
Batch boyutu	16
Öğrenme Oranı	0.0005
Giriş boyutu	3x200x200
Optimizasyon yöntemi	ADAM

Performans, sınıflandırma problemlerinde sıklıkla kullanılan doğruluk (accuracy) ve F1-ölçütü ile raporlanmaktadır. Doğruluk, modelin doğru sınıflandırdığı tüm örneklerin toplam örnek sayısına oranını vermektedir. F1-ölçütü ise modelin kesinlik (precision) ve duyarlılık (recall) değerlerinin harmonik ortalaması ile hesaplanır. Burada kesinlik, bir sınıf olarak tahmin edilen örneklerin ne kadarının doğru olduğunu, duyarlılık ise bir sınıfa ait olan örneklerin ne kadarının model tarafından doğru tahmin edildiğini gösterir. Çok sınıflı sınıflandırmada, F1 ölçütü her bir sınıf için ayrı ayrı hesaplanır ve ardından ortalama değerler alınır. Burada iki farklı yaklaşım bulunmaktadır ve bu tez çalışması kapsamında bu iki yaklaşım ile de performans raporlanmıştır. Makro-F1 ölçütü, sınıflarda yer alan veri sayısını dikkate almadan, her bir sınıfın F1 değerlerinin (c . sınıf için $F1_c$) aritmetik ortalaması ile hesaplanır. Böylece, tüm sınıflar eşit ağırlıkta değerlendirilerek, sınıflar arasındaki dengesizliklerin etkisi minimize edilir. Diğer yandan, Ağırlıklı-F1 ölçütü, her bir sınıfın veri setindeki örnek sayısına göre ağırlıklandırılmış F1 ortalama değerini vermektedir. Böylece, daha fazla örneğe sahip sınıflar, skor üzerinde daha büyük bir etkiye sahip olur. Veri sayısının düşük olmasının performansı negatif etkileyeceği göz önüne alındığında, bu metrik, özellikle sınıf dağılımının dengesiz olduğu veri setlerinde modelin performansını daha adil bir şekilde değerlendirmek için kullanılır. Tez çalışması kapsamında raporlanan doğrulama metriği, modelin doğru sınıflandırdığı tüm örneklerin oranı olarak hesaplandığından bu metrikte, sınıflardaki veri sayısının dikkate alındığı

söylenbilir. $F1_c$ c . sınıf için F1 değeri ve n_c c . sınıfın örnek sayısı olmak üzere Makro-F1 ve Ağırlıklı-F1 ölçütleri 3.1 ve 3.2’te gösterildiği gibi hesaplanmaktadır.

$$\mathbf{Makro-F1(\%)} = \frac{1}{7} \sum_{c=1}^7 (F1_c) \quad (3.1)$$

$$\mathbf{Ağırlıklı-F1(\%)} = \frac{1}{\sum_{c=1}^7 n_c} \sum_{c=1}^7 n_c (F1_c) \quad (3.2)$$

Ek olarak, CNN modelinin hesaplama maliyetini ve bellek gereksinimini belirleyen, GFlops (Giga Floating Point Operations per Second) ve parametre sayısı da raporlanmıştır. Bu değerler, modelin verimliliği ve özellikle uçta uygulanabilirliği açısından kritik öneme sahiptir. Önerilen yöntemin hızı, gerçek uygulamalarda kullanılabilirliğinin gösterilebilmesi amacıyla Raspberry Pi bilgisayarı üzerinde gerçekleştirilen testler için FPS (Frames Per Second) cinsinden raporlanmıştır.

3.2 Derin Öğrenme Modellerinin Performansa Etkisi

Tablo 3.2’de, dört farklı CNN mimarisi kullanılarak elde edilen performanslar karşılaştırmalı olarak sunulmaktadır. Bunun yanı sıra, hesaplama yoğunluğu (GFlops) ve parametre sayıları da raporlanmıştır¹. Tabloda referans model olarak, [32]’de sunulan aktivite tanıma modeli gösterilmektedir. Bu model, ResNet18 modeli ile sınıflandırma yapmakta, ancak önerilen ön işleme adımlarını uygulamamaktadır. Kısaca, 3 boyutlu spektrogramlardan, örtüşme olmaksızın oluşturulan 4 saniyelik bölümler ($w = 4, s = 4$) eğitim ve test için kullanılmıştır. Etiketleme için çalışmada ayrıntı verilmemekle birlikte, herhangi bir güven eşik parametresinden bahsedilmemektedir. Benzer modelin kullanılması nedeni ile referans[32] yöntem ve önerilen ResNet18 kullanan yöntemin karşılaştırılması, ön işleme ve iyileştirme adımları ile doğruluğun %8 oranında arttırıldığını göstermektedir. Tablo, en düşük GFlops değerine sahip olan AlexNet ile %89 doğruluk oranına ulaşıldığını göstermektedir. Bu değer, diğer modellere göre düşük olmakla birlikte, özellikle sınırlı işlemci gücüne sahip olunan cihazlar için kullanılabilirliğini göstermiştir. Bunun yanı sıra, ResNet34 ile %95.86 doğruluk performansına ulaşılmıştır. Sınıflardaki örnek sayılarının dikkate alındığı Ağırlıklı-F1 ve alınmadığı Makro-F1 metrikleri incelendiğinde, özellikle VGG16, ResNet18 ve ResNet34’in ağırlıklı ortalama benzer performanslara ulaştığı gözlemlenmektedir. Bu durum, özellikle az sayıda örnek içeren sınıflar için ResNet18’in diğer iki yöntemden daha düşük performans sergilediğini, ResNet34’ün ise sınıflar arasında daha dengeli bir

¹<https://pytorch.org/vision/main/models.html>

performansa ulaştığını göstermektedir.

Tablo 3.2 Farklı omurga ağları ve literatür yöntemleri için GFlops, parametre sayısı, doğruluk ve F1-ölçütleri

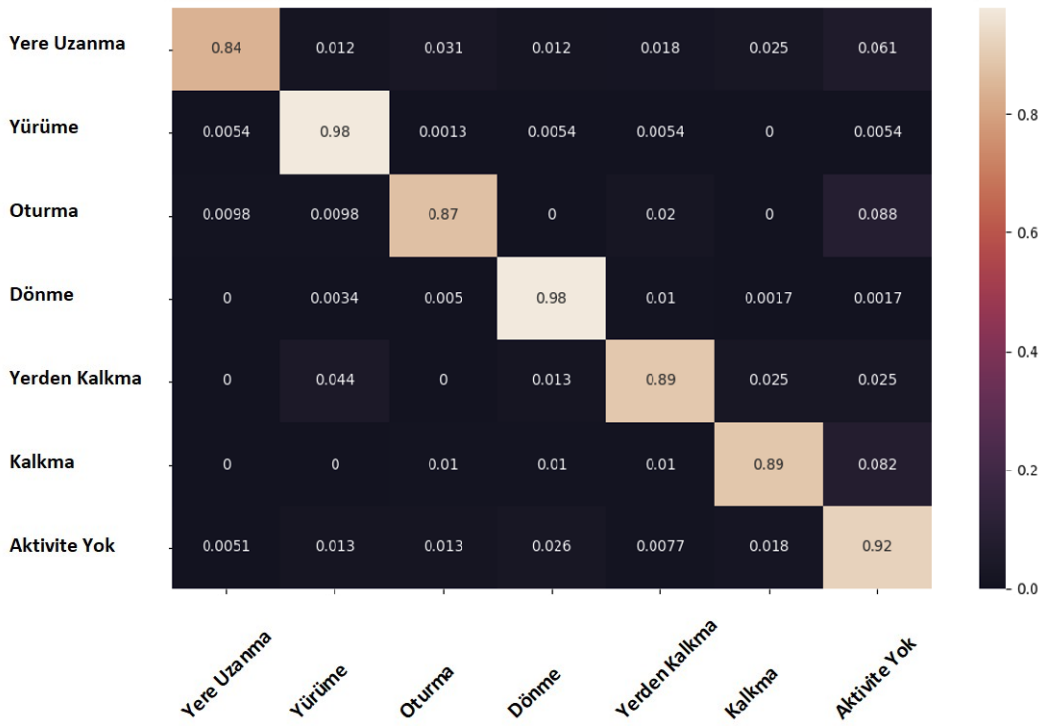
	GFlops	Parametre	Doğruluk (%)	Makro F1 (%)	Ağırlıklı F1 (%)
Ref1[32]-ResNet18	1.81	11M	86.5	-	-
Ref2[25]-FT	-	-	95.9	94.3	-
Ref2[25]-ResNet34	-	21M	96.6	94.9	-
AlexNet	0.71	57M	89.42	82.84	89.88
VGG16	15.47	134M	94.26	94.37	96.45
ResNet18	1.81	11M	94.71	92.97	96.08
ResNet34	3.66	21M	95.86	95.22	96.31

Literatürde transformatör mimarisine dayalı Füzyon Transformatör (FT) modelini öneren [25], OperaNet PWR (3 kanallı) ve kanal durum bilgisi (Channel State Information – CSI - 4 kanallı) verilerini tümleştirerek aktivite sınıflandırma gerçekleştirmektedir. Burada, her bir kanal transformatör girişine verilecek olan segment (patch) olarak kabul edilmektedir. Aynı çalışmada raporlanmış ResNet34 tabanlı modelde ise giriş verisi, 7 boyutlu görüntü olarak alınmıştır. [32] ile benzer şekilde girdiler, spektrogramlardan örtüşme olmadan 4 saniyelik bölütler ile oluşturulmuştur. Tablo 3.2 incelendiğinde [25] doğruluk oranının %95.9 olarak raporlandığı ve tez çalışmasında ResNet34 modeli ile (%95.86) benzer sonuçlara ulaşıldığı görülmektedir. Buna ek olarak, önerilen veri iyileştirme yaklaşımlarının katkısını daha iyi gözlemlemek amacıyla, [25]'de PWR ve CSI verisinin tümleştirildiği ve ResNet34 mimarisinin temel alındığı sınıflandırıcı sonuçları da raporlanmıştır. Görüldüğü üzere, [25]-ResNet34 ile, tez kapsamında önerilen ResNet34 mimarisinin kullanıldığı yöntemle göre %1'lik bir performans artışı olmaktadır. Bu artış, [25]'de ek olarak kullanılan 4 kanallı CSI verisinin doğal bir sonucudur. Tez kapsamında önerilen yaklaşımlar ile, sınırlı veri durumunda bile benzer sınıflandırma doğruluğuna ulaşılabildiği açıktır.

Veri seti üzerindeki toplam performansın yüksek olması, modelin genel doğruluğu açısından önemli olmakla birlikte, sınıf bazında dengeli bir performans sağlanması da kritik öneme sahiptir. Bu amaçla, Şekil 3.1, 3.2, 3.3 ve 3.4 de sırasıyla, AlexNet, VGG16, ResNet18 ve ResNet34 için karmaşıklık metrisleri raporlanmıştır. Karmaşıklık matrisi, modelin her bir sınıf için ne kadar doğru ve yanlış tahmin yaptığını ve böylece modelin sınıf bazlı performansını göstermektedir. Sınıflar arasında yüksek performans farklılığı, modelin her sınıfta tutarlı sonuçlar üretememesi anlamına gelmekte ve model güvenilirliğini azaltmaktadır. Ayrıca, belirli sınıfların performansının ön plana çıkması istenilen durumlarda da bu inceleme kritik öneme sahiptir.



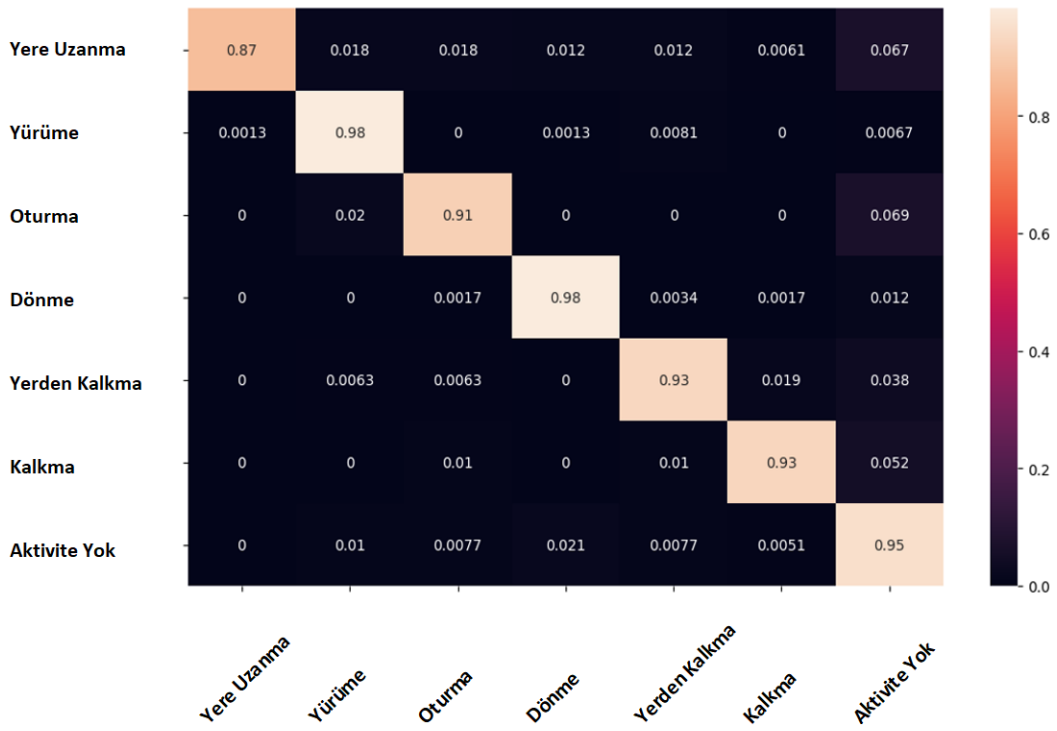
Şekil 3.1 AlexNet karmaşıklık matrisi



Şekil 3.2 VGG16 karmaşıklık matrisi



Şekil 3.3 ResNet18 karmaşıklık matrisi



Şekil 3.4 ResNet34 karmaşıklık matrisi

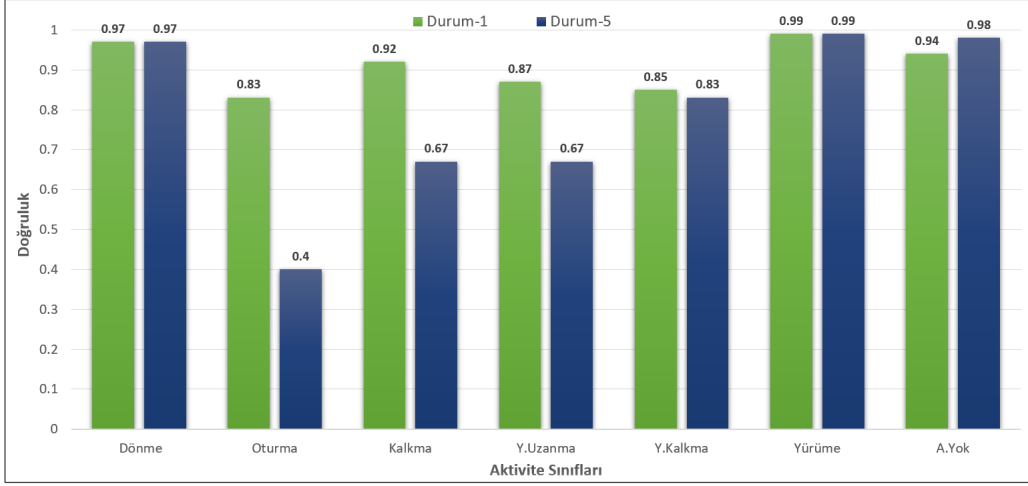
Şekiller incelendiğinde, AlexNet ve ResNet18 mimarisinde en düşük başarı oranına sahip sınıf sırasıyla %59 ve %83 doğruluk ile 'Oturma' iken, VGG16 ve ResNet34'de %84 ve %87 başarımla 'Yere Uzanma' olmuştur. AlexNet her ne kadar ortalama doğruluk değerinde kabul edilebilir bir performansa ulaşmış olsada, sınıflar arası performans dengesizlikleri açıkça görülmektedir. Ayrıca, ResNet34 doğruluk metriğine göre ResNet18'den %1'lik bir farkla öne çıksada, 7 sınıfın 6'sında %90 üstü başarıma ulaşırken, ResNet18'in 4 sınıf için %90 üstü başarımla gösterdiği görülmektedir. Önerilen veri iyileştirme yaklaşımlarının sınıf bazlı etkisini gözlemek için, referans çalışmada [32] raporlanan sonuçlar incelenmiş, %65 ile en düşük performans gösterilen 'Yerden Kalkma' sınıfı için önerilen yöntem ile %85 başarıma ulaşıldığı gözlemlenmiştir.

3.3 Yöntem Parametrelerinin Performansa Etkisi

Önerilen yöntem parametrelerinin performansa etkisini incelemek amacıyla Tablo 3.3'de 5 farklı parametre setine göre performans ve veri seti boyutu ResNet18 için raporlanmaktadır. Burada Durum-1 önerilen yöntemdir. Tüm durumlar için w bölüt süresi 4 sn olarak alınmıştır. Aktivite yok sınıfı için güven eşik değerinin $r_{ay} = 2$ olarak seçilmesi durumunda (durum-2), önerilen yöntemle göre ($r_{ay} = 3.9$ - durum-1), performansın yaklaşık %2 oranında azaldığı, eğitim süresinin ise 3 katına çıktığı raporlanmıştır. Ayrıca, veri setinde yer alacak spektrogram bölütlerinin örtüşmemesi durumunda ($s=4$, durum-3) örnek sayısının ve dolayısı ile eğitim süresinin azaldığı, ancak performansın %3 oranında düştüğü görülmüştür. Hem veri setinde yer alacak spektrogram bölütlerinin örtüşmemesi, hem de $r_{ay} = 2$ seçilmesi durumunda (durum-4) doğruluk performansının önerilen parametrelere (durum-1) göre %5 oranında düştüğü raporlanmıştır. Aktivite sınıfları ve Aktivite yok sınıfı için güven eşik değerinin $r_a = 3.9$, $r_{ay} = 3.9$ seçilmesi durumunda (durum-5) Ağırlıklı-F1 artmış ancak Makro-F1 değeri %7 oranında düşmüştür. Bu durum, Durum-5'te örnek sayısı yüksek olan sınıflarda performansın yükseldiğini, örnek sayısı az olan sınıflarda ise düştüğünü göstermektedir.

Tablo 3.3 Parametrelerin sınıflandırma performansına ve eğitim seti boyutuna etkisi, (ResNet18, $w = 4$). Durum-1, önerilen yöntemdir

	s	r_{ay}	r_a	Doğruluk (%)	Makro F1 (%)	Ağırlıklı F1 (%)	#Eğitim Seti
Durum-1	1	3.9	2	94.71	91.80	94.69	8988
Durum-2	1	2	2	92.58	82.96	92.38	18971
Durum-3	4	3.9	2	91.20	82.54	90.92	3454
Durum-4	4	2	2	89.43	73.24	88.80	4732
Durum-5	1	3.9	3.9	97.44	84.27	97.24	5148



Şekil 3.5 Önerilen yöntem ve Durum-5 için sınıf bazlı doğruluk değerleri

Durum-1 (önerilen parametreler) ve Durum-5 için sınıflara göre örnek sayıları (Tablo 3.4) ve sınıf bazlı performanslar (Şekil 3.5) incelendiğinde Durum-5 için Yürüme(0.99) ve Dönme(0.97) gibi yüksek performansa sahip sınıflar veri seti içinde yüksek oranda yer alırken, düşük performansa sahip Oturma(0.4), Kalkma(0.67), Yere Uzanma(0.67) gibi sınıfların veri sayıları çok azdır. Bu durum, Ağırlıklı-F1'a düşük başarımların etkisini azaltarak, ortalamanın yüksek görünmesine neden olmaktadır. İdealde, her sınıf için performansın dengeli olması, başka bir deyişle, Makro-F1 performans değeri ile Ağırlıklı-F1 performansının birbirine yakın olması beklenmektedir. Görüldüğü gibi, Durum-5 için bu durum tam tersidir. Durum-1 için, aktivite var olan tüm sınıflarda performansın artırıldığı veya aynı kaldığı ve sınıflar arası performansın dengeli olduğunu göstermektedir. Sonuçlar, önerilen parametrelerin, eğitim veri sayısını ve dolayısıyla eğitim süresinin sınırlı olarak artmasıyla birlikte, aktivite tanıma performansının kayde eğer oranda iyileştirildiğini açık bir şekilde göstermektedir.

Tablo 3.4 Durum-1 ve Durum-5 için eğitim ve test veri seti aktivite sınıflarına ait örnek sayıları

	#Eğitim Seti		#Test Seti	
	Durum-1	Durum-5	Durum-1	Durum-5
Dönme	2382	1158	596	290
Oturma	404	18	102	5
Kalkma	388	11	97	3
Y.Uzanma	652	85	163	21
Y.Kalkma	634	74	159	18
Yürüme	2971	2247	743	562

Farklı parametrelerin performansa etkisi incelendikten sonra, örneklerin etiketlenmesi için önerilmiş $f(r)$ fonksiyonun ve bölütler arası örtüşmenin

olmadığı durum (r_{ay} ve r_a eşik değerlerinin var olmaması ve $s = 4$) ile karşılaştırmalar yapılarak, bu modüllerin önemi gösterilmiştir. Tablo 3.5’de, önerilen yöntem ve yeni durum (durum-6) performansları tüm CNN mimarilerinde karşılaştırmalı olarak gösterilmektedir. Her iki yöntemde de bölüt uzunluğu $w = 4$ sn olarak alınmıştır. Durum-6’da eşik değerlerinin var olmaması sebebi ile bölüt içerisindeki en uzun süreli aktivite ile etiketleme yapılmıştır (süresine bakılmaksızın). Bu parametreler ile oluşturulan durum-6, referans çalışma [32] ile benzer bir yapıya sahip olmakla birlikte, eğitim-test setinin oluşturulması ve mimari hiper-parametrelerindeki farklılıklar birebir aynı performansın elde edilmesini engellemektedir. Genel olarak en düşük performansın gösterildiği AlexNet mimarisinde durum-1 ile iyileştirme Doğruluk ve Ağırlıklı-F1 için %6 dolaylarındadır. Referans çalışmada [32] kullanılan ResNet18 mimarisi için Durum-1 ve Durum-6 arasında Doğruluk değerindeki yaklaşık %10’luk artışa ek olarak Makro-F1 değerinde raporlanan %25’den fazla artış, sınıf bazlı eleman sayılarından bağımsız olarak her sınıf için başarımın artırıldığı bir göstergesidir.

Tablo 3.5 Farklı omurga ağları ile için, doğruluk ve F1-ölçütünün iki farklı parametre setinde karşılaştırılması

	Doğruluk (%)		Makro-F1 (%)		Ağırlıklı-F1 (%)	
	Durum-1	Durum-6	Durum-1	Durum-6	Durum-1	Durum-6
ResNet18	94.71	84.47	91.80	62.09	94.69	84.24
ResNet34	95.86	87.46	93.94	66.74	95.87	86.95
VGG16	94.26	85.63	91.03	63.94	94.26	85.49
AlexNet	89.42	83.89	82.94	65.34	89.22	83.84

Son olarak, önerilen yöntemin gerçek uygulamalarda kullanılabilirliğinin gösterilmesi amacıyla uçta uygulanabilirliği, Raspberry Pi mini bilgisayarda yapılan hız testleri ile gösterilmiştir. Tablo 3.6, tüm derin öğrenme modellerinin ilgili donanımdaki saniye başına kare (frame per second-fps) değerlerini göstermektedir. Hız hesaplamasına, verilerin, ön işleme ve iyileştirme ile bölütlenmiş görüntülere dönüştürülmesi ve aktivite sınıfının belirlenmesine kadar olan tüm adımlar dahil edilmiştir. Bölütlerde kaydırma işleminin $s = 1$ sn olarak gerçekleştirilmesi, her 1 saniyede aktivite tanıma yapıldığını ifade etmektedir. Bu durumda, hızın 1 fps altına inmemesi gerektiği açıktır. Çalışmada, AlexNet 15 fps ile en yüksek hıza ulaşırken, VGG16 1.2 fps ile en düşük hıza sahiptir. Bu durum, yöntemlerin sahip oldukları GFlops (Tablo 3.2) değerleri ile tutarlılık göstermektedir. Ayrıca, ResNet tabanlı her iki yöntemde 1 fps’nin çok üstünde bir hıza ulaşmıştır. Şuan için veriler, dış dünyadan bir sensör aracılığı ile alınmamakta, hafızadan okunmaktadır. Gelecekte yapılacak entegrasyon ile sensörler aracılığı ile veri anlık olarak okunabilecektir. Bu durum,

hızı etkileyecek olsada, var olan versiyonda ulařılan yüksek hızlar, entegrasyonun yöntemin istenilen hızın altında çalışmasına yol açmayacağı açıktır.

Tablo 3.6 Raspberry Pi 5 ile farklı derin öğrenme mimarileri için hız karşılaştırması

	AlexNet	VGG16	ResNet18	ResNet34
fps	15	1.2	8	5



4

SONUÇ

Bu çalışmada, pasif Wi-Fi sensörlerinden elde edilen veriler kullanarak insan aktivitelerinin tanınma doğruluğunu artırmaya yönelik farklı veri ön işleme ve iyileştirme teknikleri önerilmiştir. Önerilen yöntemler ile veri setleri oluşturulmuş ve AlexNet, VGG16, ResNet18-34 derin öğrenme mimarileri uçtan-uca eğitilerek, aktivite tanıma performansları karşılaştırmalı raporlanmıştır. ResNet18 ile elde edilen nümerik sonuçlar, sınıflandırma doğruluğunun referans çalışmaya göre %8 oranında artırıldığını göstermektedir. Performanstaki bu kayda değer artış, veri kalitesinin iyileştirmesinin doğal bir sonucudur. Ayrıca, veri etiketleme için önerilen metot ile, sınıf örneklerindeki dengesizliklerin, özellikle "aktivite yok" sınıfında sadece yüksek güvenilirlikteki örneklerin dahil edilmesi gibi yöntemlerle giderilmesi sağlanmış, böylece modelin genel performansının artırıldığı gözlemlenmiştir. Bu yaklaşım, veri seti oluşturma süreçlerinin, derin öğrenme modellerinin performansına etkisini açıkça göstermektedir.

Önerilen yöntemin gerçek uygulamalarda kullanılabilirliğin araştırılması için, testler Raspberry Pi 5 cihazında gerçekleştirilmiştir. AlexNet, 15 fps ile en yüksek hıza ulaşmış ve özellikle düşük işlem gücüne sahip cihazlarda gerçek zamanlı uygulamalar için uygun bir seçenek olmuştur. Buna ek olarak performansın AlexNet'e oranla yaklaşık %5 arttırdığı ResNet18 tabanlı yöntem ile 8 fps hızlara ulaşılmıştır. Bu yöntem hız ve doğruluk açısından dengeli bir yöntem olarak ön plana çıkmaktadır. Bu araştırmanın sonuçlarının, akıllı ev sistemleri, yaşlı veya engelli insanların hareketlerinin izlenmesi ve davranış analizi gibi pratik uygulamalar için önemli katkılar sunacağına inanılmaktadır.

Önerilen yöntemler, farklı sensörlerden elde edilen veriler ile (ultra geniş band, giyilebilir sensörler gibi) yapılacak aktivite tanıma uygulamalarında kullanılabilir. Ayrıca, aktivite tanımda olduğu gibi, zaman serisi analizinin gerekli olduğu, finansal analizler, sağlık izleme sistemleri gibi birçok uygulama için verilerin iyileştirilmesine katkı sağlayacağı düşünülmektedir.

- [1] M. B. Shaikh, D. Chai, S. M. S. Islam, N. Akhtar, “From cnns to transformers in multimodal human action recognition: A survey,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 20, no. 8, pp. 1–24, 2024. doi: 10.1145/3664815.
- [2] M. A. Khan *et al.*, “Human action recognition using fusion of multiview and deep features: An application to video surveillance,” *Multimed Tools Appl.*, vol. 83, pp. 14 885–14 911, 2024. doi: 10.1007/s11042-020-08806-9.
- [3] A. Zahin, L. T. Tan, R. Q. Hu, “Sensor-based human activity recognition for smart healthcare: A semi-supervised machine learning,” in *Artificial Intelligence for Communications and Networks*, 2019, pp. 450–472. doi: 10.1007/978-3-030-22971-9_39.
- [4] D. Liciotti, M. Bernardini, L. Romeo, E. Frontoni, “A sequential deep learning application for recognising human activities in smart homes,” *Neurocomputing*, vol. 396, pp. 501–513, 2020. doi: 10.1016/j.neucom.2018.10.104.
- [5] M. Nan *et al.*, “Human action recognition for social robots,” in *22nd International Conference on Control Systems and Computer Science*, 2019, pp. 675–681. doi: 10.1109/CSCS.2019.00121.
- [6] G. Guo, A. Lai, “A survey on still image based human action recognition,” *Pattern Recognition*, vol. 47, no. 10, pp. 3343–3361, 2014. doi: 10.1016/j.patcog.2014.04.018.
- [7] E. Ergün, F. Gürkan, O. Kaplan, B. Günsel, “Video action classification by deep learning,” in *25th Signal Processing and Communications Applications Conference*, 2017, pp. 1–4. doi: 10.1109/SIU.2017.7960446.
- [8] A. Poulouse, J. H. Kim, D. S. Han, “Hit har: Human image threshing machine for human activity recognition using deep learning models,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–21, 2022. doi: 10.1155/2022/1808990.
- [9] S. Srimath, Y. Ye, K. Sarker, R. Sunderraman, S. Ji, “Human activity recognition from rgb video streams using 1d-cnns,” in *IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Internet of People and Smart City Innovation*, 2021, pp. 295–302. doi: 10.1109/SWC50871.2021.00048.

- [10] B. X. B. Yu, Y. Liu, K. C. C. Chan, C. W. Chen, "Egcn++: A new fusion strategy for ensemble learning in skeleton-based rehabilitation exercise assessment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 9, pp. 6471–6485, 2024. doi: 10.1109/TPAMI.2024.3378753.
- [11] J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, 2019. doi: 10.1016/j.patrec.2018.02.010.
- [12] M. A. A. Al-qaness, A. Dahou, M. A. Elaziz, A. M. Helmi, "Multi-resatt: Multilevel residual network with attention for human activity recognition using wearable sensors," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 144–152, 2023. doi: 10.1109/TII.2022.3165875.
- [13] B. X. Yu, Y. Liu, X. Zhang, S. Zhong, K. C. Chan, "Mmnet: A model-based multimodal network for human action recognition in rgb-d videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3522–3538, 2023. doi: 10.1109/TPAMI.2022.3177813.
- [14] D. Liu *et al.*, "Temporal cues enhanced multimodal learning for action recognition in rgb-d videos," *Neurocomputing*, vol. 594, 2024. doi: 10.1016/j.neucom.2024.127882.
- [15] M. M. Islam, S. Nooruddin, F. Karray, "Multimodal human activity recognition for smart healthcare applications," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2022, pp. 196–203. doi: 10.1109/SMC53654.2022.9945513.
- [16] A. A. Badawi, A. Al-Kabbany, H. Shaban, "Multimodal human activity recognition from wearable inertial sensors using machine learning," in *IEEE-EMBS Conference on Biomedical Engineering and Sciences*, 2018, pp. 402–407. doi: 10.1109/IECBES.2018.8626737.
- [17] D. Girish, V. Singh, A. Ralescu, "Understanding action recognition in still images," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 1523–1529. doi: 10.1109/CVPRW50498.2020.00193.
- [18] S. R. Hosseyni, S. Seyedin, H. Taheri, *Human action recognition in still images using convit*, 2024. arXiv: 2307.08994.
- [19] P. Khaire, P. Kumar, J. Imran, "Combining cnn streams of rgb-d and skeletal data for human activity recognition," *Pattern Recognition Letters*, vol. 115, pp. 107–116, 2018. doi: 10.1016/j.patrec.2018.04.035.
- [20] A. Jain, R. Akerkar, A. Srivastava, "Privacy-preserving human activity recognition system for assisted living environments," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 5, pp. 2342–2357, 2024. doi: 10.1109/TAI.2023.3323272.
- [21] S. Arshad *et al.*, "Wi-chase: A wifi based human activity recognition system for sensorless environments," in *IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks*, 2017, pp. 1–6. doi: 10.1109/WoWMoM.2017.7974315.

- [22] W. Jiao, C. Zhang, “An efficient human activity recognition system using wifi channel state information,” *IEEE Systems Journal*, vol. 17, no. 4, pp. 6687–6690, 2023. doi: 10.1109/JSYST.2023.3293482.
- [23] P. Hu, W. Liu, C. Yang, Y. Sun, J. Li, “Wifi csi based passive human activity recognition method using blstm-cnn,” in *IEEE 23rd International Conference on Communication Technology*, 2023, pp. 210–215. doi: 10.1109/ICCT59356.2023.10419414.
- [24] Y. Ge *et al.*, “A doppler-based human activity recognition system using wifi signals,” in *IEEE Sensors*, 2021, pp. 1–4. doi: 10.1109/SENSORS47087.2021.9639680.
- [25] A. K. Koupai, M. J. Bocus, R. Santos-Rodriguez, R. J. Piechocki, R. McConville, “Self-supervised multimodal fusion transformer for passive activity recognition,” *IET Wireless Sensor Systems*, vol. 12, no. 5-6, pp. 149–160, 2022. doi: 10.1049/wss2.12044.
- [26] M. Bocus, “A comprehensive multimodal activity recognition dataset acquired from radio frequency and vision-based sensors.,” *figshare. Collection*, 2022. doi: 10.6084/m9.figshare.c.5551209.v1.
- [27] M. Muaaz, A. Chelli, A. A. Abdelgawwad, A. C. Mallofré, M. Pätzold, “Wiwehar: Multimodal human activity recognition using wi-fi and wearable sensing modalities,” *IEEE Access*, vol. 8, pp. 164 453–164 470, 2020. doi: 10.1109/ACCESS.2020.3022287.
- [28] H. Zou, J. Yang, H. P. Das, H. Liu, Y. Zhou, C. J. Spanos, “Wifi and vision multimodal learning for accurate and robust device-free human activity recognition,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 426–433. doi: 10.1109/CVPRW.2019.00056.
- [29] A. Krizhevsky, I. Sutskever, G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017. doi: 10.1145/3065386.
- [30] K. Simonyan, A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations*, 2015, pp. 1–14.
- [31] K. He, X. Zhang, S. Ren, J. Sun, “Deep residual learning for image recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
- [32] M. Bocus, V. S. Li W., et al., “Operanet, a multimodal activity recognition dataset acquired from radio frequency and vision-based sensors,” *Scientific Data*, vol. 9, pp. 1–18, 2022. doi: 10.1038/s41597-022-01573-2.
- [33] M. I. Skolnik, *Radar Handbook*. The McGraw-Hill Companies, 2008.
- [34] B. Karanfil, B. Tuysuz, “Pasif radar uygulamalarında çoklu ardışık ofdm kanalı kullanımının menzil çözünürlüğüne etkilerinin incelenmesi,” *Gümüşhane Üniversitesi Fen Bilimleri Dergisi*, vol. 10, no. 4, pp. 1049–1056, 2020. doi: 10.17714/gumusfenbil.508017.

- [35] W. Li, R. J. Piechocki, K. Woodbridge, C. Tang, K. Chetty, “Passive wifi radar for human sensing using a stand-alone access point,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 59, no. 3, pp. 1986–1998, 2021. doi: 10.1109/TGRS.2020.3006387.
- [36] S. Ioffe, C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015, pp. 448–456.
- [37] D. P. Kingma, J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations*, 2015.
- [38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.



TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirisi

1. S. Gölcük, F. Gürkan and N. Kahraman, "Enhancing Passive Wi-Fi Sensor Data for Improved Activity Recognition," *2024 32nd Signal Processing and Communications Applications Conference (SIU)*, Mersin, Türkiye, 2024, pp. 1-4, doi: 10.1109/SIU61531.2024.10600954.

