

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ TEZLİ YÜKSEK
LİSANS PROGRAMI**

TINYML TABANLI GÖRSEL İŞİTSEL ANAHTAR KELİME TESPİTİ

HAZIRLAYAN

MEHMET TOSUN

YÜKSEK LİSANS TEZİ

ANKARA-2024

**BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ TEZLİ YÜKSEK
LİSANS PROGRAMI**

TINYML TABANLI GÖRSEL İŞİTSEL ANAHTAR KELİME TESPİTİ

HAZIRLAYAN

MEHMET TOSUN

YÜKSEK LİSANS TEZİ

PROF. DR. HAMİT ERDEM

ANKARA-2024

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Elektrik Elektronik Mühendisliği Anabilim Dalı Elektrik Elektronik Mühendisliği Tezli Yüksek Programı çerçevesinde Mehmet TOSUN tarafından hazırlanan bu çalışma, aşağıdaki jüri tarafından Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 19/07/2024

Tez Adı: TinyML Tabanlı Görsel İşitsel Anahtar Kelime Tespiti

Tez Jüri Üyeleri (Unvanı, Adı- Soyadı, Kurumu)	İmza
Doç. Dr. Derya Yılmaz, Gazi Üniversitesi	
Prof. Dr. Hamit Erdem, Başkent Üniversitesi	
Doç. Dr. Selda Güney, Başkent Üniversitesi	

ONAY

Prof. Dr. DİLEK ÇÖKELİLER SERDAROĞLU

Fen Bilimleri Enstitüsü Müdürü

Tarih:/...../2024

BAŞKENT ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih: 25/07/ 2024

Öğrencinin Adı, Soyadı: Mehmet TOSUN

Öğrencinin Numarası: 22020373

Anabilim Dalı: Elektrik – Elektronik Mühendisliği Anabilim Dalı

Programı: Tezli Yüksek Lisans Programı

Danışmanın Unvanı/Adı, Soyadı: Prof. Dr. Hamit Erdem

Tez Başlığı: TinyML Tabanlı Görsel İşitsel Anahtar Kelime Tespiti

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 81 sayfalık kısmına ilişkin, 25/07/2024 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı %8’dir. Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimeden daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası:

ONAY

Öğrenci Danışmanı

Prof. Dr. Hamit ERDEM

Tarih: 25/07/2024

TEŞEKKÜR

Yüksek lisans öğrenim hayatım boyunca göstermiş olduğu tüm anlayış, sabır ve hoşgörüsü olmak üzere, çalışmanın sonuca ulaştırılmasında her zaman yardımcı ve yol gösterici olduğu için tez danışmanım Prof. Dr. Hamit Erdem'e,

Veri seti oluşturmada bana destek olan FNSS ailesinden EKMELE AYANOĞLU, BURAK KEMAL KARA, MEHMET KEMAL TANKUT, YAVUZ AKŞAN, YUNUS ATLI, AZİZ ÜÇER ve diğer arkadaşlarıma,

Eğitim hayatımda desteğini esirgemeyen kıymetli aileme,

Son olarak, tezimin değerlendirilmesinde katkıda bulunan jüri üyelerine, zamanlarını ayırdıkları ve değerli geri bildirimleri için teşekkür ederim.

Mehmet TOSUN

ÖZET

Mehmet TOSUN

TinyML TABANLI GÖRSEL İŞİTSEL ANAHTAR KELİME TESPİTİ

Başkent Üniversitesi Fen Bilimleri Enstitüsü

Elektrik Elektronik Mühendisliği Anabilim Dalı

2024

Anahtar kelime tespiti (AKT), makine öğreniminin kullanıldığı alanlardan birisidir. Amacı, ses veya görüntü verisinden belirli kelime veya objenin otomatik tespit edilmesidir. Taşınabilir yapay zekâ uygulamalarının artmasıyla beraber, bu alanda da uygulamalar artmaktadır. Özellikle AKT uygulamalarının etkinliğini artırmak için hibrit sistemler (ses ve görüntünün birlikte kullanımı) üzerinde çalışma yapılmaktadır. Bu sistem ile birlikte iki farklı kanaldan algılanan ses ve görüntü komutlarının birleştirilmesi amaçlanmaktadır. Bilgisayar (PC) ortamında görsel işitsel AKT üzerinde birçok çalışma yapılmış ve iyi sonuçlar elde edilmiştir. Diğer taraftan derin öğrenme uygulamalarını düşük kapasiteli işlemciler üzerinde gerçekleştirmek için TinyML (Düşük Kapasiteli Makine Öğrenmesi) kapsamında çalışmalar yapılmaktadır. Bu uygulamalarda, derin öğrenmeye yönelik geliştirilen modelin parametrelerini azaltarak (nicelleştirme, kırpma) sıradan mikrodenetleyici üzerinde uygulama imkânı oluşturmaktadır. Bu çalışmada ses ve görüntü verisi kullanılarak, TinyML alanında AKT uygulaması önerilmiştir. Önerilen hibrit modelin eğitiminde öncelikle ses ve görüntü modelleri Edge Impulse yazılım ortamında ayrı ayrı eğitilmiştir. Geliştirilen MobileNetV2 ve CNN tabanlı modeller ESP32-CAM ve Arduino Nano BLE geliştirme kitlerine yüklenerek, denenmiştir. Daha sonra modeller doğrusal ağırlıklı birleştirme metodu ile birleştirilerek denenmiştir. Sistemin başarısı standart ölçütlere göre test edilmiştir. Deneysel sonuçlarda doğruluk ölçütüne göre, sadece ses tabanlı AKT başarısı %85, sadece görüntü tabanlı AKT başarısı %85 olurken, görsel işitsel hibrit uygulamasında sınıflandırma başarısı %90 civarında olmuştur.

ANAHTAR KELİMELER: TinyML, Makine öğrenimi, Anahtar kelime tespiti, Evrimsel Sinir ağıları, Edge impulse

ABSTRACT

Mehmet TOSUN

TinyML-BASED VISUAL AUDIO KEYWORD DETECTION

Baskent University Institute of Science

Department of Electrical and Electronics Engineering

2024

Keyword detection (KWD) is one of the areas where machine learning is used. Its purpose is the automatic detection of specific words or objects from audio or image data. As portable artificial intelligence applications become more prevalent, the number of applications in this field is also growing. In particular, hybrid systems (the use of audio and video together) are being studied to increase the effectiveness of KWD applications. The system aims to combine audio and visual commands detected through two different channels. Extensive work has been done on audiovisual keyword detection in a computer environment, yielding good results. On the other hand, efforts are being made within the scope of TinyML (Low-Power Machine Learning) to implement deep learning applications on low-capacity processors. In these applications, reducing the parameters of the deep learning model (quantization, pruning) makes it possible to implement the model on ordinary microcontrollers. In this study, a keyword detection application in the field of TinyML is proposed using audio and visual data. In the training of the proposed hybrid model, the audio and visual models were first trained separately in the Edge Impulse software environment. Developed MobileNetV2 and CNN-based models were loaded onto ESP32-CAM and Arduino Nano BLE development kits and tested. Subsequently, the models were combined using a linear weighted fusion method and tested. In the experimental results, according to the accuracy criterion, the success rate of the audio-based KWD was 85%, the success rate of the image-based KWD was 85%, while the classification success in the audiovisual hybrid application was around 90%.

KEYWORDS: TinyML, Machine learning, Keyword detection, Convolutional neural networks, Edge impulse

İÇİNDEKİLER

TEŞEKKÜR.....	i
ÖZET	ii
ABSTRACT	iii
İÇİNDEKİLER.....	iv
TABLO LİSTESİ.....	vi
ŞEKİLLER LİSTESİ	vii
SİMGELER VE KISALTMALAR.....	ix
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	3
2.1. Anahtar Kelime Tespiti.....	5
2.2. Tezin Amacı	7
2.3. Tezin Yapısı.....	7
3. MAKİNE ÖĞRENİMİ	9
3.1. Makine Öğrenimi Türleri	9
3.1.1. Gözetimli öğrenme	9
3.1.2. Gözetimsiz öğrenme	10
3.1.3. Yarı gözetimli öğrenme.....	11
3.1.4. Güçlendirilmiş öğrenme.....	11
3.2. Makine Öğrenimi Algoritmaları	12
3.2.1. Lojistik regresyon.....	12
3.2.2. Destek vektör makineleri.....	13
3.3. Makine Öğreniminin Kullanım Alanları	14
4. TinyML	15
4.1. TinyML Neden Tercih Edilmelidir	15
4.2. TinyML Nasıl Çalışır	16
4.2.1. TinyML’de veri toplama.....	17
4.2.2. TinyML’de eğitim.....	17
4.2.3. TinyML’de optimizasyon	18

4.3. TinyML Mimarileri.....	21
4.3.1. Evrişimsel sinir ağları (CNN).....	21
4.3.2. Yinelemeli sinir ağı.....	26
4.3.3. MobileNets	27
4.4. TinyML Araç Setleri.....	31
4.4.1. Donanım	32
4.4.2. Yazılım.....	33
5. TinyML İLE GÖRSEL İŞİTSEL ANAHTAR KELİME TESPİTİ	35
5.1. Görsel Anahtar Kelime Tespiti	35
5.1.1. Görsel AKT’de veri toplama	36
5.1.2. Görsel AKT’de dürtü tasarımı.....	36
5.1.3. Görsel AKT’de eğitim	40
5.1.4. Görsel AKT’de dağıtım ve nicelme.....	42
5.2. İşitsel Anahtar Kelime Tespiti.....	43
5.2.1. İşitsel AKT’de veri toplama	44
5.2.2. İşitsel AKT’de dürtü tasarımı	45
5.2.3. İşitsel AKT’de eğitim	47
5.2.4. İşitsel AKT’de dağıtım	47
5.3. Görsel İşitsel Anahtar Kelime Tespiti	48
6. DENEYLER VE TARTIŞMA	52
6.1. İşitsel AKT Deneyleri.....	53
6.1.1. İşitsel AKT’de veri seti büyüklüğü etkisi	53
6.1.2. İşitsel AKT’de farklı insan seslerinin etkisi	54
6.1.3. İşitsel AKT’de ortam gürültüsü etkisi.....	55
6.2. Görsel AKT Deneyleri.....	56
6.2.1. Görsel AKT’de veri boyutları etkisi	56
6.2.2. Görsel AKT’de ortam aydınlığı etkisi	56
6.2.3. Görsel AKT’de farklı şekildeki ellerin etkisi	57
6.3. Görsel İşitsel AKT Deneyleri.....	58
6.3.1. Görsel İşitsel AKT’de ortam aydınlığının etkisi.....	58
6.3.2. Görsel işitsel AKT’de formül katsayılarının etkisi	58
7. SONUÇLAR.....	60
KAYNAKLAR.....	61

TABLO LİSTESİ

	Sayfa
Tablo 4.1. TinyML destekli donanımlar	32
Tablo 5.1. Görsel AKT’de kullanılan eğitim parametreleri	40
Tablo 5.2. Görsel AKT’de optimizasyon sonucu Edge Impulse tarafından hesaplanan değerler	42
Tablo 5.3. Farklı çalışmalarda kullanılan Yes- No veri setinin doğruluk değerleri.....	44
Tablo 5.4. İşitsel sınıflandırmada eğitim parametreleri	47
Tablo 5.5. İşitsel AKT’de optimizasyon sonucu Edge Impulse tarafından hesaplanan değerler	47
Tablo 6.1. Open-Close veri seti büyüklüğünün doğruluğa etkisi.....	54
Tablo 6.2. Yes-No veri seti büyüklüğünün doğruluğa etkisi	54
Tablo 6.3. Farklı insan seslerinin test doğruluğuna etkisi	55
Tablo 6.4. Ortam gürültüsünün test doğruluğuna etkisi	55
Tablo 6.5. Görsel veri boyutlarının doğruluk, RAM ve çıkarım zamanına etkisi.....	56
Tablo 6.6. Görsel AKT’de ortam aydınlığının test doğruluğuna etkisi	56
Tablo 6.7. Görsel AKT’de ortam aydınlığının test doğruluğuna etkisi	57
Tablo 6.8. Görsel işitsel AKT’de ortam aydınlığının etkisi	58
Tablo 6.9. Görsel işitsel AKT’de farklı katsayıların etkisi.....	59

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 1.1. Yapay zekanın alt bileşenleri.....	1
Şekil 2.1. Anahtar kelime tespitinde ön işleme.....	5
Şekil 2.2. Anahtar kelime tespitinde konuşma tanıma.....	6
Şekil 3.1. Makine öğrenimi türleri.....	9
Şekil 3.2. Gözetimli öğrenme bloğu	10
Şekil 3.3. Gözetimsiz öğrenme bloğu	11
Şekil 3.4. Karar ağaçları algoritması örneği	13
Şekil 3.5. Makine öğreniminin başlıca kullanım alanları	14
Şekil 4.1. TinyML ile model oluşturmadaki akış şeması.....	17
Şekil 4.2. TinyML’de veri toplama işlemi.....	17
Şekil 4.3. TinyML’de kullanılan optimizasyon türleri	19
Şekil 4.4. TinyML’de nicelleştirme işlemi	20
Şekil 4.5. TinyML’de budama işlemi	21
Şekil 4.6. CNN sinir ağı mimarisi.....	22
Şekil 4.7. CNN’de evrişim işlemi	22
Şekil 4.8. Havuzlama işlemi	23
Şekil 4.9. ReLu aktivasyon fonksiyon grafiği	24
Şekil 4.10. Sigmoid aktivasyon fonksiyonu grafiği.....	25
Şekil 4.11. Tanh aktivasyon fonksiyon grafiği	25
Şekil 4.12. RNN modeli çalışma senaryosu	26
Şekil 4.13. Standart konvolüsyon işlemi	28
Şekil 4.14. Derin ayrıştırma konvolüsyon işlemi.....	28
Şekil 4.15. Noktasal evrişim işlemi	29
Şekil 4.16. Edge Impulse ile TensorFlow Lite kıyaslaması.....	34

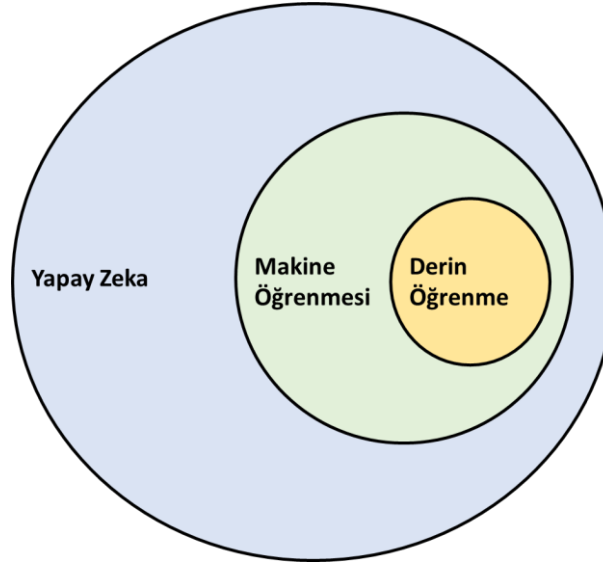
Şekil 5.1. Edge Impulse platformunda model oluşturma adımları	35
Şekil 5.2. Açık el hareketi (A) ve kapalı el hareketi (B).....	36
Şekil 5.3. Edge Impulse platformundaki veri boyutlandırma modları.....	37
Şekil 5.4. Görüntü sınıflandırmada kullanılan işleme blokları	37
Şekil 5.5. Görüntü sınıflandırmada kullanılan öğrenme blokları	38
Şekil 5.6. Transfer Learning mimarileri	38
Şekil 5.7. Classification mimarileri	39
Şekil 5.8. Görsel sınıflandırma sonucu elde edilen özellik gezgini.....	39
Şekil 5.9. Görsel AKT eğitimi sonrası Edge Impulse’da hesaplanmış karışıklık matrisi. 41	
Şekil 5.10. Arduino IDE’de ESP32 konfigürasyon ayarlaması.....	42
Şekil 5.11. UART haberleşmesinin şematik gösterimi	43
Şekil 5.12. Görsel AKT modelinin el hareketlerini sınıflandırması	43
Şekil 5.13. Veri seti oluştururken konuşmacıların mikrofona uzaklığı	45
Şekil 5.14. MFCC yapısının blok şeması	45
Şekil 5.15. İşitsel veri zaman genlik grafiği (A) ve MFCC spektrogramı (B).....	46
Şekil 5.16. İşitsel sınıflandırma sonucu oluşan özellik gezgini	46
Şekil 5.17. İşitsel AKT eğitimi sonrası Edge Impulse’da hesaplanmış karışıklık matrisi.. 47	
Şekil 5.18. İşitsel AKT modelinin sesleri sınıflandırması	48
Şekil 5.19. Görsel işitsel AKT blok şeması	48
Şekil 5.20. JAVA tabanlı tasarlanan grafiksel arayüz	49
Şekil 5.21. GUI ortamında görsel işitsel AKT modelinin çalıştırılması.....	50
Şekil 5.22. GUI ortamında AKT sonuçları	50
Şekil 5.23. Donanım ortamında görsel işitsel AKT	51
Şekil 6.1. Çalışmada yapılan deneyler.....	53
Şekil 6.2. Görsel AKT testlerinde kullanılan eller.....	57

SİMGELER VE KISALTMALAR

AKT	Anahtar Kelime Tespiti
AWS	Amazon Web Servisi
BLE	Bluetooth Düşük Enerji
CNN	Evrişimsel Sinir Ağları
CPU	Merkezi İşlem Birimi
DL	Derin Öğrenme
DSP	Dijital Sinyal İşleme
FFT	Hızlı Fourier Dönüşümü
GRU	Kapılı Rekürrent Birimler
GUI	Grafiksel Arayüz
LSTM	Uzun Kısa Süreli Bellek
MFCC	Mel Frekansı Kepstrum Katsayıları
ML	Makine Öğrenimi
MSE	Ortalama Karesel Hata
NLP	Doğal Dil İşleme
NN	Sinir Ağları
RNN	Yinelemeli Sinir Ağları
SNR	Oransal Gürültü Azaltma
SVM	Destek Vektör Makineleri
TFLM	Tensorflow Lite Micro
TinyML	Küçük Makine Öğrenimi
UART	Evrensel Asenkron Alıcı Verici
WPE	Ağırlıklı Tahmin Hatası

1. GİRİŞ

Yapay zekâ (YZ), son yıllarda teknolojik ilerlemenin önde gelen alanları arasında yer almaktadır. Hızla gelişen bu alan, birçok sektörde uygulanmaya başlanmış ve insan hayatını önemli ölçüde etkilemeye başlamıştır. YZ, insan zekasını taklit eden ve belirli görevleri yerine getirebilen bilgisayar sistemleri ve yazılımların geliştirilmesini içeren bir bilim dalıdır [1]. YZ'nın temel çalışma prensibi, büyük miktarda veriyi işleyebilen ve bu verilerden öğrenebilen algoritmaların kullanımına dayanır [2]. Bu sistemler, verileri analiz ederek, desenleri tanıyarak ve karar verme süreçlerini simüle ederek çalışırlar. YZ'nın tarihi, 1950'lerde Alan Turing'in "Makineler düşünebilir mi?" sorusunu ortaya atmasıyla başlar ve bu soru, YZ'nın felsefi temellerini oluşturur. YZ, büyük verileri işleyip analiz ederek öğrenir ve bu bileşenleri gelecekteki kararlarında kullanır. Bu algoritmik süreçler, makine öğrenimi ve derin öğrenme gibi yöntemleri içermektedir. Şekil 1.1 Hata! Başvuru kaynağı bulunamadı.'de yapay zekanın alt bileşenleri gösterilmiştir.



Şekil 1.1. Yapay zekanın alt bileşenleri

Makine öğrenimi (ML), PC sistemlerinin deneyimlerinden öğrenmeye ve gelişimine odaklanarak problem ve görevleri; tahminler ve tespitler yoluyla gerçekleştirmesidir [3,4]. İnsanların çok karmaşık ve geniş veri kümesi üzerinde doğru sonuçları çıkarabilme ve geleceği tahmin edebilme yeteneklerini otomatikleştiren bir teknolojidir. ML birçok açıdan önemlidir ve modern toplumda birçok alanda büyük etkilere sahiptir. Bu teknoloji, birçok endüstride inovasyon ve gelişmeyi tetikleyebilir, insan yaşamını iyileştirebilir ve iş

süreçlerini optimize edebilir. Bunun en önemli sebeplerinden derin öğrenme (DL) tekniğiyle birçok bilimsel sorunu çözme yeteneğine sahip olmasıdır [5].

DL, yapay sinir ağları kullanarak veriden anlamlı bilgiler çıkarmayı amaçlayan bir makine öğrenmesi alt dalıdır [6]. DL modelleri, çok katmanlı yapıları sayesinde karmaşık veri kalıplarını öğrenebilir ve yakalayabilirler. Bu modellerin temel yapı taşları olan yapay sinir hücreleri (nöronlar), birbirleriyle bağlantılı katmanlar halinde düzenlenmiştir. Girdi katmanından başlamak üzere, veriler ardışık gizli katmanlardan geçerek çıktı katmanına ulaşır. Her bir katman, verinin belirli özelliklerini öğrenir ve bir sonraki katmana aktarır. Bu süreçte, modelin performansını optimize etmek amacıyla, hata geri yayılımı (backpropagation) algoritması kullanılarak ağırlıklar güncellenir. DL, büyük veri kümeleri ve yüksek işlem gücü gerektiren uygulamalarda, özellikle görüntü ve ses tanıma, doğal dil işleme gibi alanlarda son derece başarılı sonuçlar elde etmektedir.

Son yıllarda, ML yöntemleri hızla gelişmiş ve birçok yeni uygulama alanı ortaya çıkmıştır. Bu gelişmeler, günlük yaşamımızın pek çok alanında büyük yenilikler getirmiştir. Özellikle, düşük güç tüketimi ve sınırlı donanım kaynakları ile çalışan cihazlarda ML modellerinin uygulanması, yeni fırsatlar yaratmıştır. IoT (Nesnelerin İnterneti) cihazları, giyilebilir teknoloji ve akıllı sensörler gibi alanlarda ML modellerinin entegre edilmesi, daha verimli ve otonom sistemler geliştirilmesine olanak tanımaktadır. Bu yöntemler ile, gerçek zamanlı veri analizi ve karar verme süreçleri daha hızlı ve etkili bir şekilde gerçekleştirilebilmektedir. Bu yöntemlerin biri de TinyML'dir.

TinyML, makine öğrenmesi modellerinin mikro-kontrolörler dahil olmak üzere düşük işlemci gücüne ve hafızaya sahip cihazlarda çalıştırılmasını ifade eder [7]. Genellikle 1 miliwatt veya daha az enerji tüketimi olan cihazlarda kullanılan bu modeller, her yerde bulunan cihazları ve fiziksel dünyayı daha akıllı hale getirebilir [8]. Bu sayede, akıllı sensörler, giyilebilir cihazlar ve diğer IoT ürünleri daha akıllı hale gelebilmektedir. TinyML, geleceğin bağlantılı cihazlar dünyasında önemli bir rol oynayacak gibi görünmektedir. Literatürde yeni bir yaklaşım olan TinyML teknolojisi birçok araştırmacı tarafından da incelenmektedir.

2. LİTERATÜR TARAMASI

Rashmi ve Yelchuri (2022), sınırlı kaynaklara sahip olan gömülü cihazlara makine öğrenim algoritmalarını dahil etmek için kullanılan TinyML temelleri üzerine çalışma yapmışlardır. Çalışmada TinyML'in kullanım durumları, uygulama alanları ve TinyML uygulamaları oluşturma zorlukları bahsedilmiştir. Sonuç olarak, birkaç belirli endüstriyel alan için TinyML'in katkıları ve dezavantajları belirtilmiştir [9]. Partha (2021), teknolojik gelişmelerin çok önemli bir parçası olan TinyML ile ilgili sezgisel bir inceleme yapmıştır. TinyML teknolojisinin çalışmasının arka planı, araç setleri, zorlukları ve geleceğe yönelik öneriler sunmuştur [10]. Ishan ve arkadaşları (2023), çalışmalarında yeniden düzenlenebilir TinyML çözümleri üzerinde bir araştırma sunmuşlardır. Cihaza dağıtılan TinyML modelinin daha sonrasında tekrardan güncellenmenin önemi üzerinde tartışılmıştır. Her bir hiyerarşik katmanın yeniden düzenlenebilirliği üzerinde incelemeler yapılmıştır. Sonuç olarak gömülü sistemlerin ayırt edici özellikleri ve bu cihazlara uygulanan yeniden biçimlendirilebilir çıkarım mekanizması belirtilmiştir [11]. Norah ve Dina (2022), TinyML teknolojisine genel bir bakış ve çalışmalarının incelemesi üzerine bir tartışma yapmışlardır. Çalışmalarındaki ana katkı olarak TinyML çalışmalarında kullanılan ML modellerinin türünün analizini sağlamaktır [12]. Soro (2021), çalışmada genel olarak TinyML alanının etkili kullanımı için başlıca zorlukları ve teknolojik kolaylıkları belirtip, tinyml teknolojisine genel bakış yapmıştır [13]. Schizas ve arkadaşları (2022), çalışmalarında TinyML teknolojisine genel bakış yapıp, avantaj ve dezavantajları, Tensorflow Lite çerçevesini tanımlamışlardır. Ayrıca 5G ve LPWAN ağ teknolojisine entegrasyonu hakkında araştırmalarını açıklamışlardır [14]. Boz ve Durgun (2023), Parkinson hastalığındaki insan sağlığını tehdit eden titremenin tespitini TinyML ile sağlamışlardır. Model eğitimi için Edge Impulse yazılımı, donanım olarak Arduino 33 Ble sense ve LSM9DS1 ivme sensörü kullanmışlardır. Kullanılan veri setinde istenilen titreme ve istenmeyen titreme birbirinden ayrılmıştır. Önerilen sistemin %85 tanıma doğruluğu sağladığı bulunmuştur [15]. Mansoureh (2021), çalışmada çamaşır makinesinin arıza tespiti için TinyML kullanmıştır. Donanım olarak Arduino Nano 33 BLE kullanmıştır ve Arduino bir çamaşır makinesine montaj edilmiştir. Kartta bulunan ivme sensörü ile çamaşır makinesi verileri ile arıza tespit çalışması yapmıştır. Eğitim modeli olarak CNN kullanmıştır. Çalışmada %92'lik başarı elde etmiştir [16]. GRAU (2021), çalışmada TinyML teknolojisine genel bir bakış yapıp, Arduino 33 Ble ile anahtar kelime tespiti yapmıştır. Çalışmada yes ve no işitsel verileri ile anahtar kelime tespiti üzerinde

çalışmıştır. Eğitim modeli olarak CNN kullanmıştır. Sonuç olarak model %94 eğitim doğruluğu elde edilmiştir [17]. Altayeb ve arkadaşları (2023), çalışmalarında Edge Impulse platformunu kullanarak bir SiPM mikro radyasyon sensöründe TinyML ile otomatik bir gama radyasyon algılama ve tanımlama sistemi tasarlamışlardır. Çalışmalarında kullanılan model, gerçek gama kaynağı verileri kullanılarak eğitilmiştir. Eğitim modeli olarak CNN kullanılmıştır ve eğitim Edge Impulse platformunda yapılmıştır. Yapılan testler sonucunda gerçek zamanlı işlemede %90 doğruluk elde etmişlerdir [18]. Giovanni ve arkadaşları (2023), veri seti hazırlama aşamasından başlayarak TinyML tabanlı bir model geliştirip, TinyML teknolojisinin sınırlarını göstermişlerdir. Toplanan veri seti, çeşitli ML algoritmaları ile eğitilmiştir. Eğitilen modeller; kesinlik, f1 puanı ve doğruluk ölçüm metotlarıyla karşılaştırılmıştır. Sonuç olarak %94'lük doğruluk ile en iyi sonucu Destek Vektör Makineleri ile almışlardır [19]. Azevedo ve arkadaşları (2023), belirli bir alandan geçen bireylerin hareketsiz görüntülerini işleyerek maskelerini tespit etmek için TinyML tabanlı bir yaklaşım sunmuşlardır. Bu yaklaşım, Edge Impulse platformu tarafından desteklenen MobileNet modelini referans alarak bir transfer öğrenme (learning) işlemi gerçekleştirerek tasarlanmıştır. Edge Impulse ile model Arduino Nano 33 BLE sense kartına aktarılıp OV7675 kamera modülü ile test edilmiştir. Sonuç olarak, uygun fiyatlı ve enerji tasarrufu sağlayarak, yüz maskesi algılama sistemlerinin gerçek dünyada kullanılmasına yönelik önemli bir katkı sağlanmıştır [20]. Ali Reza ve arkadaşları (2022), güneş enerjisi çiftliklerindeki panellerin anomali tespiti için kenar tabanlı bir derin öğrenme modeli kullanmışlardır. Üç farklı uç cihazda bir destek vektör makineleri sınıflandırıcı ile birlikte hem TFLite hem de TensorRT formatlarında bir Siyam ikiz kodlayıcı kullanmışlardır. Sonuç olarak ortaya çıkan enerji tüketimi ve CPU kullanımı, böyle bir sistemin, doğru anomali tespiti sağlamada Nvidia Jetson Nano donanımı en iyi şekilde dağıtılabileceğini gösterilmiştir [21]. David ve arkadaşları (2021), çalışmalarında TensorFlow Lite teknolojisi hakkında genel bir bakış olabilecek bir araştırma sunmuşlardır. TensorFlow Lite Micro (TFLM)'nin arka planındaki tasarım hakkında bilgi verilmiş, kaynak gereksinimi ve çalışma süresi verimliliğini değerlendirmişlerdir. Sonuç olarak TFLM'nin makine öğrenimini yalnızca birkaç kilobayt belleğe sahip gömülü cihazlarda etkili ve verimli bir şekilde çalıştırılabileceğini belirtmişlerdir [22].

2.1. Anahtar Kelime Tespiti

TinyML teknolojisinin kullanım alanlarından birisi de anahtar kelime tespit işlemidir. Anahtar kelime tespiti, konuşulan cümleleri veya metin parçalarını analiz ederek belirli anahtar kelimeleri otomatik olarak algılayan bir uygulamadır. Temel amacı, ses veya görüntü verisinden belirli kelime veya objenin otomatik tespit edilmesidir [23]. Taşınabilir yapay zekâ uygulamalarının artmasıyla beraber, bu alanda da uygulamalar artmaktadır. AKT genellikle metin sınıflandırma, duyum analizi, özetleme ve bilgi çıkarma gibi NLP (doğal dil işleme) görevlerinde kullanılır. AKT birkaç temel adımdan oluşur.

Ön İşleme: Metindeki gereksiz veya anlamlı olmayan kısımların temizlenmesi ve normalizasyonu işlemlerini kapsamaktadır. Ön işleme adımlarının amacı, ham metinden anlamlı ve ayırt edici kelimeleri çıkarmaktır. Şekil 2.1’de Anahtar kelime tespitindeki ön işleme adımları gösterilmiştir.

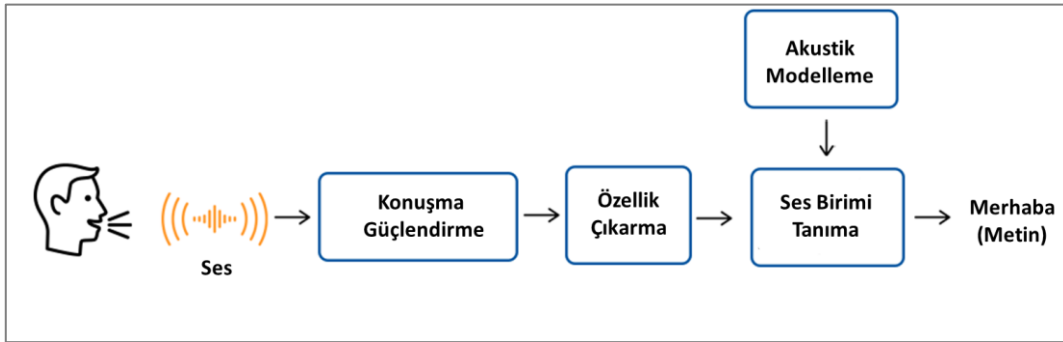


Şekil 2.1. Anahtar kelime tespitinde ön işleme

Gürültü azaltma, arka plan gürültüsünün (rüzgâr, trafik sesi) ve ortam gürültüsünün (sohbet eden insanlar, ekipman gürültüsü) azaltma işlemidir [24]. Gürültü azaltma teknikleri olarak filtreleme ve oransal gürültü azaltma (SNR) kullanılır. Filtreleme işlemi, belirli frekans aralıklarındaki gürültüyü bastırmak için frekans seçici filtrelerin kullanılmasıdır. SNR, istenen sinyalin gürültüye oranını artırmak için ortalama kare hata (MSE) gibi ölçütlerin kullanılmasıdır. Konuşma segmentasyonu, ses aktivite tespiti ve ses olmayan bölümleri (işitselsizlik, müzik) kaldırma gibi işlemler uygulanan kısımdır. Burada işitsel parçalardaki konuşma bölümlerinin belirlenmesi işlemi yapılır. İşitsel özellik çıkarma işlemi, konuşma sinyallerinden Mel Frekans Cepheleri (MFCC) gibi akustik özellikler çıkarılma işlemidir. Normalleştirme, farklı konuşma ortamlarından gelen verilerin özelliklerinin standartlaştırılma işlemidir. Farklı kişilerin İşitselleri, konuşma hızları ve

aksanları önemli ölçüde değişiklik gösterebilir. Farklı ortamlarda da gürültü seviyesi ve işitselin akustik özellikleri değişebilir. Normalleştirme, bu gibi farklılıkları azaltarak verilerin daha tutarlı hale gelmesini sağlar.

Konuşma Tanıma (Speech Recognition): İşitsel verilerindeki konuşma, metin tabanlı bir formata dönüştürülür. Bu adım, konuşmanın tanınması ve metne dönüştürülmesi için konuşma tanıma tekniklerini içerir. Şekil 2.2’de konuşma tanıma işleminin blok şeması gösterilmiştir.



Şekil 2.2. Anahtar kelime tespitinde konuşma tanıma

Anahtar Kelime Tespiti: Konuşma tanıma sonucunda elde edilen metin verileri üzerinde anahtar kelime tespiti yapılır. Bu adım, belirli anahtar kelimelerin veya ifadelerin tanınması ve belirlenmesini içerir.

Önceki çalışmalar genellikle işitsel tabanlı anahtar kelime tespiti üzerine odaklanmıştır [25,26]. Ancak bu yaklaşımlar gürültülü ortamlarda başarı sağlamada zorluklar yaşatabilmektedir. Bu sebeple AKT uygulamalarının etkinliğini artırmak için hibrit sistemler üzerinde çalışma yapılmaktadır. Bu sistem ile birlikte iki farklı kanaldan algılanan ses ve görüntü komutlarının birleştirilmesi amaçlanmaktadır.

Görsel işitsel anahtar kelime tespiti hem görüntü hem de işitsel verilerini birlikte analiz ederek, sadece işitsel verilerine dayanan geleneksel anahtar kelime tespitinden daha yüksek doğruluk ve güvenilirlik sağlamaktadır. Görsel işitsel anahtar kelime tespiti uygulamaları PC tabanlı başarılı uygulamalar olmasına rağmen, mikrodenetleyici tabanlı TinyML uygulamalarında yapılmamıştır.

2.2. Tezin Amacı

Bu tez çalışması TinyML ile gürültülü ortamlarda performansı düşen işitsel anahtar kelime tespiti için görsel verilerden de yararlanarak görsel işitsel anahtar kelime yapıp gürültülü ortamlarda da yüksek performans almayı hedeflemektedir. Çalışmada sırasıyla aşağıdaki işlemler uygulanmıştır.

- İşitsel ve görsel ifadeler için ayrı ayrı veri seti kullanılmıştır. İşitsel verileri için Yes-No, görsel veriler için Açık El- Kapalı El veri seti kullanılmıştır. Yes işitsel verisi, açık el hareketi ile eşleştirilirken, No işitsel verisi, kapalı el hareketi ile eşleştirilmiştir.
- İşitsel ve görsel modelleri Edge Impulse uygulama programında ayrı ayrı eğitilerek cihazlara dağıtımı sağlanmıştır. Ayrıca modeller cihaza dağıtılmadan önce model boyutu küçültülmesine yönelik nicelleştirme işlemi uygulanmıştır. Ardından modellerin cihazlarda ayrı ayrı başarıları test edilmiştir.
- Hibrit uygulamanın başarısını izlemek için PC ortamında GUI geliştirilmiştir. GUI ile işitsel ve görüntü komutlarını eş zamanlı olarak alarak, doğrusal ağırlıklı birleştirme yapılmıştır. Ardından standart ölçütlere göre birleştirilen model test edilmiştir. Ayrıca sistemin çalışması GUI'den bağımsız bir şekilde de cihazlar üzerinde çalıştırılmış ve sınıflandırma sonucu bir LED ile gösterilmiştir.

2.3. Tezin Yapısı

Bu tez, giriş bölümü ile birlikte 5 bölümden oluşmaktadır.

İkinci bölümde, makine öğrenimi kavramı ve bu alandaki temel prensipler detaylı bir şekilde ele alınmıştır. Bu bölümde, makine öğreniminin tanımı, uygulama alanları ve çeşitli algoritmalar üzerinde durulmuştur. Ayrıca, denetimli ve denetimsiz öğrenme gibi temel öğrenme türleri, model değerlendirme yöntemleri ve makine öğrenimi süreçlerinin adımları hakkında kapsamlı bilgiler sunulmuştur. Bu bağlamda, bölüm, okuyuculara makine öğreniminin teorik temellerini sağlam bir şekilde kavrama ve bu alandaki güncel gelişmeleri anlama fırsatı sağlamaktadır.

Üçüncü bölümde, TinyML kavramı ve bu alandaki temel prensipler detaylı bir şekilde ele alınmıştır. Bu bölümde, TinyML'nin tanımı, avantajları ve çeşitli uygulama alanları üzerinde durulmuştur. Ayrıca, TinyML için gerekli olan donanım ve yazılım bileşenleri, enerji verimliliği ve performans kriterleri gibi konular da kapsamlı bir şekilde incelenmiştir. Bu bağlamda, bölüm, okuyuculara TinyML'nin teorik temellerini sağlam bir şekilde kavrama ve bu alandaki yenilikleri anlama fırsatı sağlamaktadır.

Dördüncü bölümde, TinyML ile görsel işitsel anahtar kelime tespiti anlatılmıştır. Bu bölümde, görsel ve işitsel sınıflandırma modellerinin veri toplama aşamasından cihaza dağıtım aşamasına kadar açıklanmıştır. Daha sonrasında bu 2 modelin doğrusal ağırlıklı birleştirilmesi açıklanmıştır.

Beşinci bölümde ise yapılan deneyler karşılaştırılmış, çalışmanın literatürdeki sorunlara çözümü ve gelecekteki durumu tartışılmıştır.

3. MAKİNE ÖĞRENİMİ

Makine öğrenimi, yapay zekâ alanının bir alt dalı olarak, algoritmalar ve istatistiksel modeller yardımıyla bilgisayar sistemlerinin deneyimlerden öğrenmesini ve gözlemlerden bilgi edinmesini sağlayan bir bilim dalıdır. [27]. Bu süreç, insan müdahalesi olmaksızın veri analizi yapabilme ve karar verme yeteneği kazandırır.

Makine öğreniminin temelleri, 1950'lerde ve 1960'larda, Arthur Samuel'in bilgisayarların satranç oynamasını sağlayan programları ve Frank Rosenblatt'ın Perceptron adını verdiği ilk yapay sinir ağını geliştirmesiyle atılmıştır. O zamandan bu yana, makine öğrenimi, veri madenciliği, doğal dil işleme, görüntü tanıma ve robotik gibi birçok alanda uygulama bulmuştur. Makine öğrenimi, sürekli gelişen ve evrilen bir alan olup, gelecekte teknolojinin ve toplumun şekillenmesinde kritik bir etkiye sahip olacaktır.

3.1. Makine Öğrenimi Türleri

Makine öğrenmesi türleri, algoritmaların nasıl öğrendiği ve veriyi nasıl işlediği temelinde ayrılmıştır. Bu türler, algoritmaların öğrenme yaklaşımları ve kullanım senaryolarına dayalı olarak farklılaşır. Şekil 3.1'de makine öğrenimi türleri gösterilmiştir.

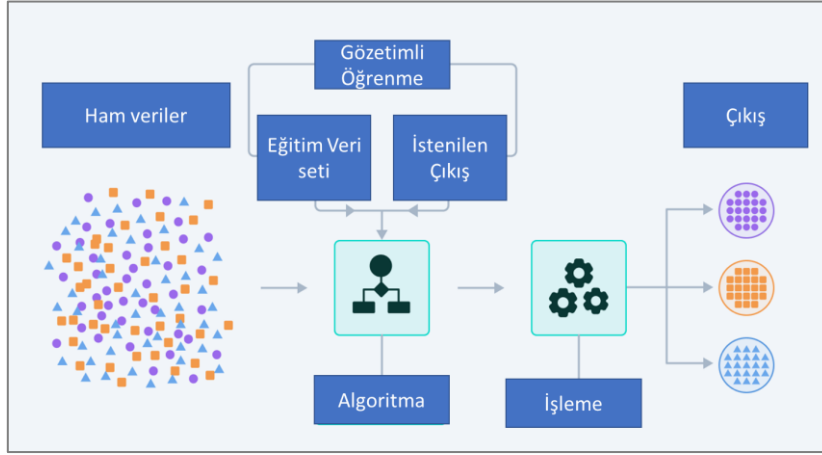


Şekil 3.1. Makine öğrenimi türleri

3.1.1. Gözetimli öğrenme

Gözetimli öğrenme, makine öğrenmesi modelinin giriş ve çıkış verilerini kullanarak öğrendiği bir türdür [28]. Adından da anlaşılacağı gibi, modelin eğitimi etiketli verilerle olmaktadır. Geliştirilen model, yeni, önceden görülmemiş verilerin çıktısını tahmin etmek

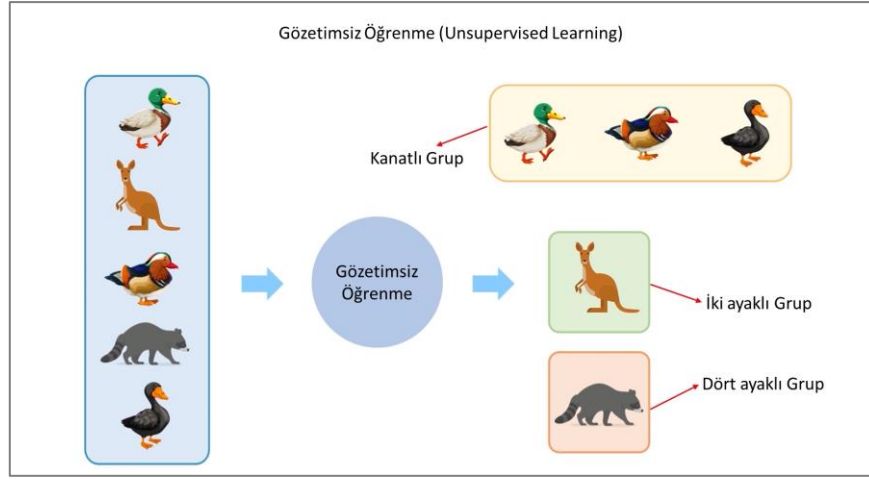
için kullanılır. Bu öğrenme türü, sınıflandırma ve regresyon problemlerini çözmek için yaygın olarak kullanılır. Şekil 3.2’de gözetimli öğrenme modeli gösterilmiştir.



Şekil 3.2. Gözetimli öğrenme bloğu [29]

3.1.2. Gözetimsiz öğrenme

Bu tür öğrenmede, etiketlenmemiş veri kümeleri, insan müdahalesine ihtiyaç duyulmadan analiz edilir [30]. Gözetimsiz öğrenme, algoritmanın içsel yapıyı keşfetmesi ve anlamlı bilgi çıkarabilmesi için verilerin dağılımını ve örüntülerini anlamaya dayanır. Bu öğrenme türünde, algoritmanın belirli bir çıktı etiketi olmadan eğitim verisi kümesinin özelliklerine dayanarak giriş verilerindeki yapıyı ya da ilişkileri kavraması beklenir. Gözetimsiz öğrenmenin yaygın uygulamaları arasında kümeleme, öneri sistemleri, yüksek boyutlu verilerin düşük boyutlu gösterimlere dönüştürülmesi ve anomalilerin tespiti bulunmaktadır. Şekil 3.3’de gözetimsiz öğrenme bloğu gösterilmiştir.



Şekil 3.3. Gözetimsiz öğrenme bloğu [31]

3.1.3. Yarı gözetimli öğrenme

Yarı gözetimli öğrenme, özellikle büyük veri kümelerinin ele alındığı durumlarda kullanılan makine öğrenmesi yaklaşımıdır. Bu teknik hem gözetimli hem de gözetimsiz öğrenme yöntemlerinin bir kombinasyonunu kullanır. Yarı gözetimli öğrenme, etiketleme maliyetlerini azaltırken aynı zamanda daha büyük bir veri setinden öğrenmenin avantajını sağlar [32]. Bu yöntem, etiketleme maliyetinin yüksek olduğu veya yeterli etiketli verinin bulunmadığı durumlarda oldukça faydalıdır.

3.1.4. Güçlendirilmiş öğrenme

Güçlendirilmiş öğrenme, makine öğrenmesinin bir dalıdır ve bir ajanın ödülü maksimize etmek suretiyle bir problemin çözümünü öğrenmeye odaklanır. Temel fikir, bir çevrede belirli eylemler gerçekleştirerek ve bu eylemler sonucunda alınan ödüllere veya cezalara dayalı olarak çevre hakkında bilgi edinmektir. Bu öğrenme türü, bir ajanın belirli bir amaç doğrultusunda belirli eylemler gerçekleştirmeyi öğrendiği bir dizi deneme yanılma süreci olarak görülebilir. Hedef, toplam ödülün maksimize edildiği bir politika belirlenmesidir.

3.2. Makine Öğrenimi Algoritmaları

Makine öğrenimi algoritmaları, PC'lerin veri analizini kullanarak kendiliğinden öğrenmesi ve geliştirmesini sağlayan bir dizi yöntemdir. Makine öğrenimi algoritmaları, farklı veri türleri ve görevler için kullanılabilir.

- **Lineer Regresyon:** Makine öğrenmesi ve istatistiksel modellemenin en temel tekniklerinden biridir. Lineer regresyon, veriye en iyi uyan doğruyu bulmaya çalışır. “En iyi uyan” terimi genellikle, modelin tahminlerinin gerçek değerlerden ne kadar farklı olduğunu ölçen bir hata fonksiyonunu minimize etmek anlamına gelir.

$$y = ax + b \quad (3.1)$$

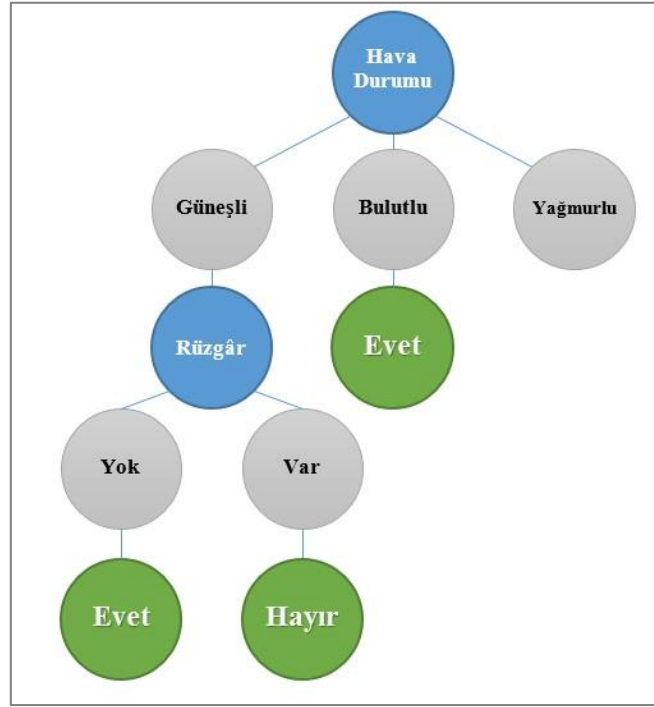
Eşitlik 3.1'deki a eğimi temsil ederken, b doğrunun y -kesişen noktasıdır. Eşitlik 3.1 sonucu olan y terimi bağımlı değişken terimini temsil ederken, x bağımsız değişkenin değeridir.

3.2.1. Lojistik regresyon

Sınıflandırma problemlerinde kullanılan bir makine öğrenimi algoritmasıdır. Genellikle ikili sınıflandırma problemlerinde yani sonuç iki olası sonuçtan birine ait olduğunda kullanılır. Lojistik regresyonun amacı, belirli bir sınıfa ait olma olasılığını veren bir model oluşturmaktır. Bu belirli bir veri noktasının belirli bir sınıfa ait olma olasılığını tahmin etmek için kullanılabilir. Lojistik regresyon, çıktısı olarak olasılık ürettiği için lineer regresyondan farklıdır. Söz konusu olasılık, belirli bir sınıfa ait olma olasılığıdır ve bu ilk başta 0 ile 1 arasında bir değer alır. Ancak, genellikle bir eşik değeri belirlenir (örneğin 0.5) ve bu eşik değerine göre bir sınıflandırma yapılır. Olasılık eşikten büyükse, veri noktası bir sınıfa, eşikten küçükse diğer sınıfa atanır.

- **Karar Ağaçları:** Karar ağaçları, özellikle sınıflandırma ve regresyon için kullanılır ve tıpkı isminden de anlaşıldığı gibi, ağaç yapısında kararları modelleyip tahmin yapar [33]. Her ağaç düğümler, dallar ve yapraklardan oluşmaktadır. Düğümler, bir öznitelik veya bir özellik seçer ve veriyi belirli bir eşiği temel alarak ikiye bölmektedir. Dallar, düğümden çıkan her bir dal için belirtilen özniteliğe göre verinin

bir alt kümesini temsil eder. Yapraklar, karar ağacının en uç bölümünde yer alırlar ve verinin son sınıfını, yani tahmini çıktıyı temsil etmektedir. Şekil 3.4’de karar ağaçları yapısının bir örneği gösterilmiştir.



Şekil 3.4. Karar ağaçları algoritması örneği

3.2.2. Destek vektör makineleri

Makine öğreniminde destek vektör makineleri (SVM), sınıflandırma ve regresyon problemleri için kullanılan bir tür gözetimli öğrenme algoritmasıdır [34]. SVM, eğitim verilerindeki herhangi bir noktadan en uzak olan iki sınıf arasında bir karar sınırı bulmaya çalışır. Bu karar sınırı, her bir sınıftaki verileri mümkün olduğunca iyi ayıracak şekilde seçilir. SVM’ler hem doğrusal hem de doğrusal olmayan veriler için kullanılabilir. Doğrusal veriler için, SVM bir doğru veya hiper düzlem kullanarak sınıfları ayırabilir. Doğrusal olmayan veriler için, SVM, Kernel dönüşümleri kullanarak doğrusal bir uzaya dönüştürülebilir ve ardından doğrusal bir SVM kullanılarak sınıflandırılabilir.

$$w * x + b = 0 \quad (3.2)$$

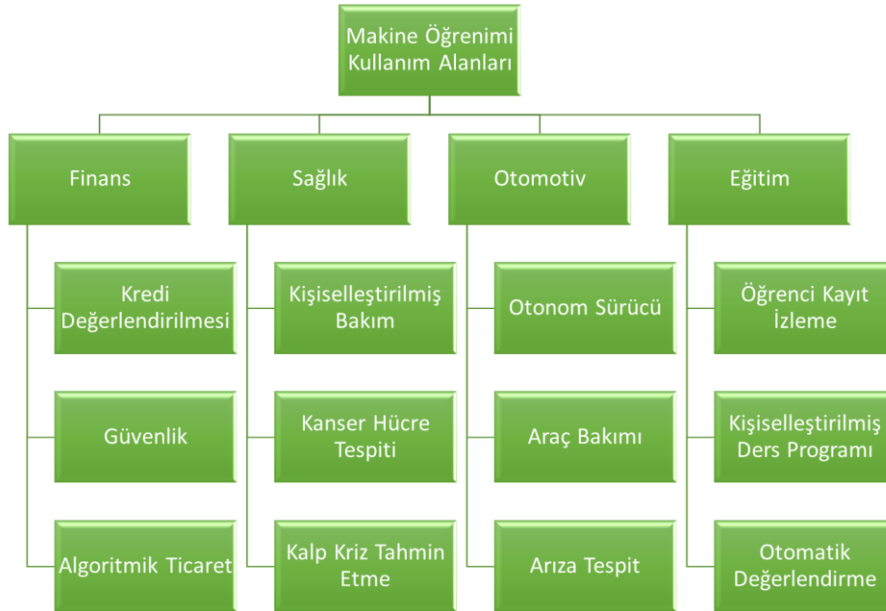
Eşitlik 3.2’de w hiper düzlemi tanımlayan ağırlık vektörüdür, x bir veri noktasıdır ve b hiper düzlemdeki sabit bir kaymadır. Sınıflandırma kararı için Eşitlik 3.3 kullanılır.

$$\text{sign}(w * x + b) \quad (3.3)$$

Eşitlik 3.3’de belirtilen denklemin sonucu pozitif olursa veri noktası birinci sınıfa, negatif bir değer elde edilirse ikinci sınıfa ait olmaktadır.

3.3. Makine Öğreniminin Kullanım Alanları

ML, PC bilimleri ve yapay zekâ alanlarında hızla gelişen ve pek çok endüstride devrim yaratan önemli bir teknolojidir. Bu teknoloji, sağlık hizmetlerinden finansal piyasalara, perakende sektöründen otomotiv endüstrisine kadar geniş bir yelpazede uygulama bulmaktadır. Şekil 3.5’de ML’nin başlıca kullanım alanları gösterilmektedir.



Şekil 3.5. Makine öğreniminin başlıca kullanım alanları

4. TinyML

TinyML “Tiny Machine Learning” teriminin kısaltmasıdır ve genellikle düşük güçte çalışan mikrodenetleyiciler üzerinde makine öğrenimi modellerinin çalıştırılmasını ve yürütülmesi anlamına gelir. Geleneksel makine öğrenimi uygulamaları genellikle yüksek güçlü PC’ler veya bulut tabanlı hizmetler üzerinde gerçekleştirilirken, TinyML, IoT cihazları gibi kaynak kısıtlı cihazlarda çalışacak şekilde optimize edilmiştir.

TinyML, mikrodenetleyicilerde çalışacak şekilde ölçeklendirilmiş hafif ve enerji-verimli makine öğrenimi algoritmalarını içerir. Bu algoritmalar genellikle küçük boyutlu ve düşük bellek kullanımı gerektirir, böylece cihazın sınırlı kaynaklarıyla uyumlu hale gelir. Örneğin, bir akıllı termostatın sıcaklık verilerini analiz edip kullanıcı tercihlerine göre ısıtma sistemini ayarlaması gibi basit ancak etkili görevler için TinyML kullanılabilir.

4.1. TinyML Neden Tercih Edilmelidir

TinyML, yapay zekâ ve makine öğrenimi modellerinin ultra düşük güç tüketimiyle çalışabilen donanım, algoritmalar ve yazılımları kapsayan hızla büyüyen bir alanı ifade eder. Bu teknoloji, özellikle pil ile çalışan cihazlarda sürekli açık kullanım durumlarını mümkün kılarak, IoT cihazlarda oldukça faydalıdır.

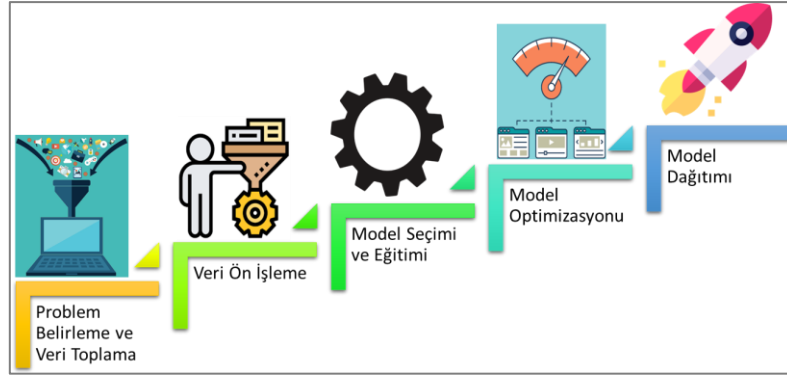
TinyML'nin önemi, IoT cihazlarının sayısının hızla artması ve bu cihazların ürettiği veri miktarının yönetilmesi gerekliliği ile daha da belirginleşmektedir. TinyML, bu verileri etkin bir şekilde işleyerek, daha akıllı ve bağlantılı bir dünya yaratma yolunda önemli bir adım olarak görülmektedir. TinyML'nin gelişimi, makine öğrenimi ve yapay zekâ teknolojilerinin geleceğini şekillendirecek ve düşük güçlü cihazlarda zekanın yaygınlaşmasını sağlayacak önemli bir faktördür. Bu nedenle, TinyML'nin tercih edilmesi, teknolojinin sürdürülebilir ve erişilebilir bir geleceğe doğru ilerlemesinde kritik bir rol oynamaktadır. TinyML teknolojisinin avantajları aşağıda sıralanmıştır:

- **Düşük Güç Tüketimi:** TinyML gömülü cihazlar üzerinde düşük güç tüketimi gerektiren uygulamalarda kullanılmak üzere optimize edilmiştir. Bu durum, batarya gücüyle çalışan cihazlar için önemlidir ve pil ömrünü uzatabilir.
- **Gizlilik ve Güvenlik:** TinyML modelleri, veriyi cihazlarda işlemektedir. Bu cihazlar internete bağımsız bir şekilde veriyi işlemektedir. Bu durum veri gizliliği ve güvenliği açısından avantaj sağlamaktadır.
- **Maliyet:** TinyML, küçük ve ucuz mikrodenetleyiciler veya gömülü sistemlerde çalışmayı sağlar. Bu tür cihazlar, genellikle düşük maliyetlidir, bu durum TinyML tabanlı çözümleri daha erişilebilir bir duruma getirmektedir.
- **Hızlı Çalışma:** TinyML modelleri, düşük güçlü cihazlarda bile hızlı bir şekilde çalışmayı sağlar. Özellikle gerçek zamanlı uygulamalarda bu durum çok önemlidir. Hızlı cevap süresi gereksinimi olan sistemlerde etkin bir şekilde desteklenebilir.
- **Kenar İşleme:** TinyML, veri analizini ve işlemeyi cihazlarda gerçekleştirdiği bir kenar hesaplama yaklaşımını benimsemektedir. Bu, veri merkezlerine veya bulut tabanlı hizmetlere bağımlılığı azaltır.
- **Esneklik:** TinyML, geniş bir cihaz yelpazesinde çalışabilir ve çeşitli uygulamalarla birlikte çalışabilir. Bu, IoT cihazlarından akıllı sensörlere ve taşınabilir cihazlara kadar birçok farklı endüstriyel ve tüketici uygulaması için potansiyel sunmaktadır.

4.2. TinyML Nasıl Çalışır

TinyML'nin çalışma prensibi, genellikle standart ML uygulamaları ile benzerdir fakat daha düşük işlem gücü, hafıza ve enerji kaynaklarına sahip olan cihazlardaki bir uygulama için özel olarak optimize edilmiştir. Geleneksel ML modelleri genellikle büyük ölçekli PC'lerde veya sunucularda çalışırken, TinyML modelleri daha az kaynak gerektiren cihazlarda çalışır.

TinyML, özel olarak optimize edilmiş makine öğrenimi algoritmaları ve sıkıştırma teknikleri kullanır. Bu teknikler, ML modellerinin boyutunu ve hesaplama gereksinimlerini azaltırken, performansını korur. Bu optimizasyonlar sayesinde, TinyML modelleri küçük mikrodenetleyicilerde bile çalışabilir. Şekil 4.1'de TinyML akış şeması gösterilmektedir.



Şekil 4.1. TinyML ile model oluşturmadaki akış şeması

4.2.1. TinyML’de veri toplama

TinyML’de veri toplama, makine öğrenimi modelini eğitmek için gerekli olan verilerin toplanması işlemidir. Bu süreç genellikle sensörlerden, mobil cihazlardan, IoT cihazlarından ve diğer benzeri platformlardan veri çekmeyi içerir. TinyML modelinin performansı büyük ölçüde toplanan bu verinin kalitesine, miktarına ve temsiliyetine bağlıdır. Toplanan verinin modelin beklentilerini karşılayacak şekilde etiketlenmesi ve ön belleği de bu sürecin bir parçasıdır. Şekil 4.2’de TinyML modelindeki veri toplama akış şeması gösterilmiştir.



Şekil 4.2. TinyML’de veri toplama işlemi

4.2.2. TinyML’de eğitim

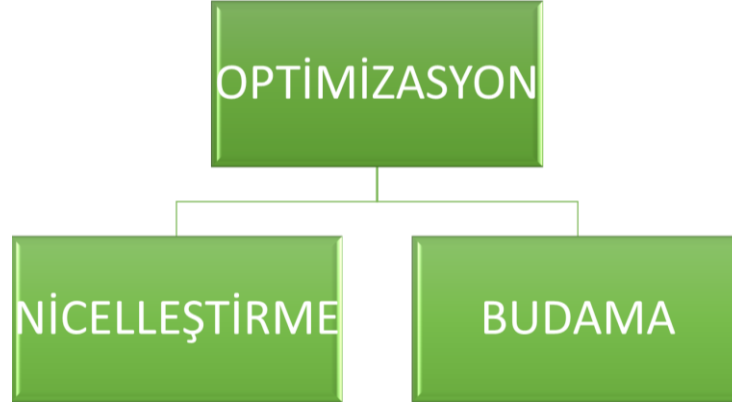
TinyML’de eğitim süreci, büyük ölçekli makine öğrenimi uygulamalarından farklıdır. Eğitim işlemi genellikle yüksek performanslı bir PC’de gerçekleştirir ve daha sonra bu eğitilmiş model, bir mikrodenetleyici gibi bir hedef cihazda çalıştırılmak üzere optimize edilir. Eğitim aşamasında, model bir dizi eğitim örneğini alır ve her bir örneğin gerçek ve tahmin edilen çıktıları arasındaki hata oranını hesaplar. Ardından, bu hatayı azaltmak için modelin ağırlıklarını günceller. Bu işlem, belirlenen hata düzeyine ulaşana veya belirli bir

sayıda iterasyon tamamlanana kadar tekrar edilir. TinyML’de eğitim süreci, genellikle iki aşamadan oluşur.

- **Modelin geliştirilmesi:** Bu aşamada, modelin mimarisi, hiper parametreleri ve kullanılacak veri seti belirlenir. Modelin geliştirilmesi, genellikle yüksek performanslı bir PC veya bulut ortamında yapılır. Modelin geliştirilmesinde, modelin doğruluğu kadar boyutu, hafıza kullanımı ve işlem süresi gibi metrikler de önemlidir. Çünkü modelin küçük cihazlarda çalışabilmesi için bu metriklerin optimize edilmesi gerekir.
- **Modelin küçültülmesi:** Bu aşamada, modelin boyutu, hafıza kullanımı ve işlem süresi gibi metrikler küçültülür. Modelin küçültülmesi, genellikle modelin sıkıştırılması, budanması, nicelemesi veya bilgi distilasyonu gibi teknikler kullanılarak yapılır. Modelin küçültülmesinde, modelin doğruluğunun çok fazla düşmemesi amaçlanır. Modelin küçültülmesi sonucunda, model küçük cihazlarda çalışabilecek hale gelir.

4.2.3. TinyML’de optimizasyon

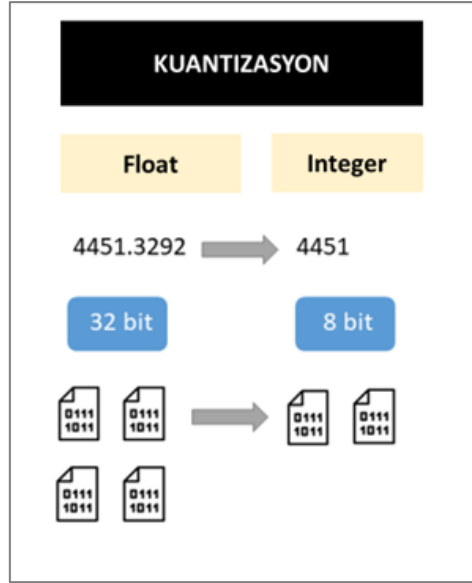
TinyML, düşük güç tüketimi, sınırlı bellek ve işlem kapasitesi gibi zorluklarla karşı karşıyadır. Bu nedenle, TinyML’de optimizasyon süreci çok önemlidir. Optimizasyon süreci, TinyML modelinin performansını ve verimliliğini artırmak için yapılan bir dizi adımdan oluşur. TinyML’de optimizasyon işlemi, bir makine öğrenmesi modelinin hem boyutunu hem de hesaplama gereksinimlerini azaltmayı hedefler. Bu kısıtlamalar çoğunlukla modelin hafıza (RAM ve Flash) kullanımı, işlemci hızı (CPU) ve enerji tüketimi ile ilgilidir. Bu üç kısıtlama, doğrudan makine öğrenme modelinin karmaşıklığı ve boyutu ile orantılıdır. Şekil 4.3’de TinyML’de kullanılan optimizasyon türleri gösterilmiştir.



Şekil 4.3. TinyML’de kullanılan optimizasyon türleri

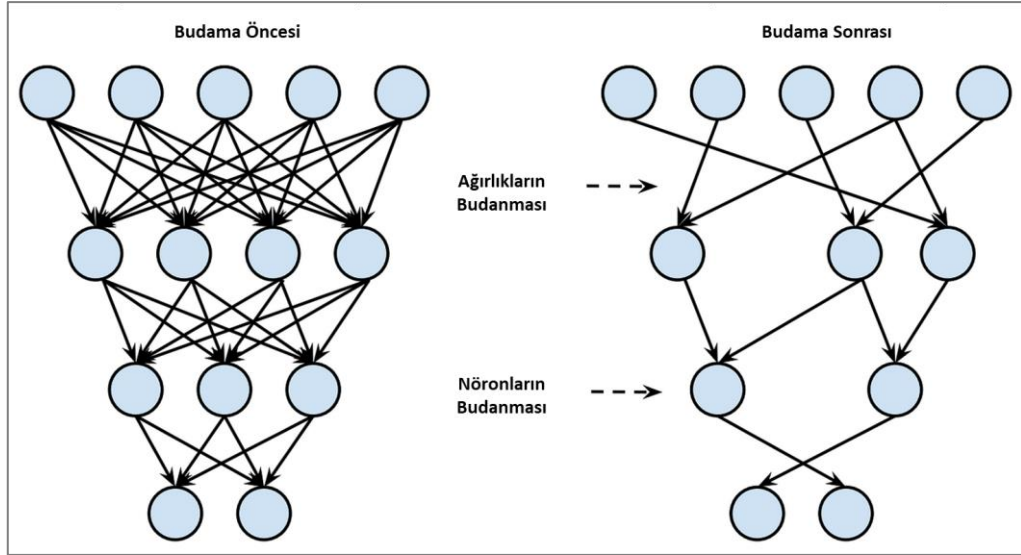
- Nicelleştirme: Nicelleştirme (quantization), TinyML’de genellikle modelin boyutunu küçültmek ve hızını artırmak için kullanılan bir tekniktir [35]. Spesifik olarak olarak, modelin ağırlıklarının hassasiyetini azaltarak ve daha az dahili hafıza kullanarak çalışmasına olanak tanır.

Büyük veri merkezlerinde çalışırken, model ağırlıkları genellikle 32 bitlik kayan nokta değerleri (float32) olarak saklanır. Ancak bu, donanımsal olarak düşük cihazlarda (örneğin cep telefonu veya mikrokontrolcüde) gereğinden fazla hassasiyet olabilir ve gereksiz yer kaplar. Nicelleştirme, bu ağırlıkları daha düşük hassasiyetli bir biçime (örneğin int8 veya uint8) dönüştürerek modelin hafızada kapladığı alanı önemli ölçüde azaltabilir [36]. Ayrıca nicelleştirme, genellikle hesaplama hızını da artırır, çünkü daha düşük hassasiyetli işlemler genellikle hızlıdır ve az hafıza kullanır. Bu durum, daha enerji harcayan ve daha hızlı yanıt veren bir model oluşturmaya yardımcı olur. Şekil 4.4’de TinyML’deki niceleme işlemi gösterilmiştir.



Şekil 4.4. TinyML’de nicelleştirme işlemi

- Budama: Budama, modelin boyutunu küçültmeye ve bellek kullanımını azaltmaya yardımcı olur [37]. Buradaki amaç, bir sinir ağının karmaşıklığını azaltmak ve modelin boyutunu küçültmek için modelden bazı “gereksiz” sinir bağlantılarını veya ağırlıklarını çıkarmaktır [38]. TinyML modelleri için özellikle faydalıdır çünkü bellek ve işlemci gücü sınırlıdır. Model boyutunu azaltmak ve işleme gereksinimlerini azaltmak, bu tür cihazlarda daha verimli ve hızlı çalışmayı sağlar. Ancak, budama işlemi dikkatlice yapılmazsa, modelin performansında kayıplara neden olabilir, bu nedenle dikkatli bir denge sağlamak önemlidir. Budama yöntemi, genellikle bir eşik değeri belirlenerek gerçekleştirilir. Eşik değerinden düşük olan parametre veya ağırlıklar budanır, yani modelden çıkarılır. Bu şekilde, ağın gereksiz veya düşük katkı sağlayan kısımları çıkarılarak modelin boyutu küçültülür. Şekil 4.5’de budama işlemi gösterilmiştir.



Şekil 4.5. TinyML’de budama işlemi [39]

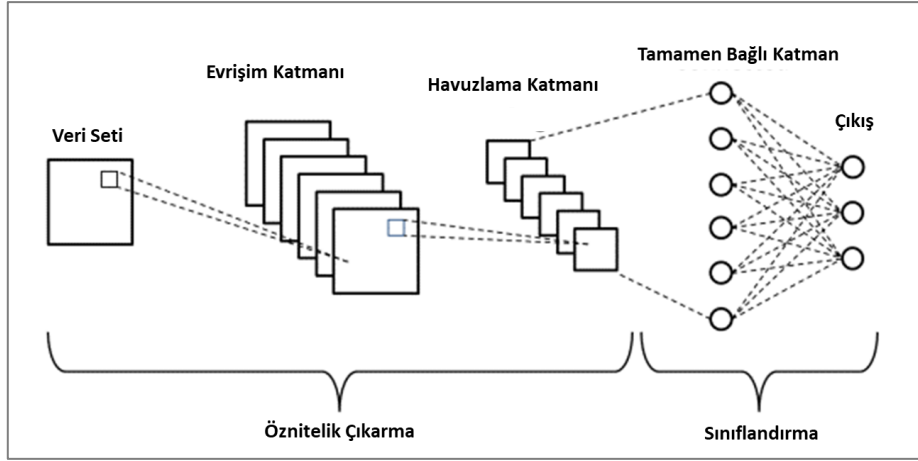
4.3. TinyML Mimarileri

TinyML için kullanılan sinir ağı mimarileri, küçük mikrodenetleyicilerde çalışabilmesi için optimize edilmiş hafif ve verimli yapıları içerir.

4.3.1. Evrimsel sinir ağları (CNN)

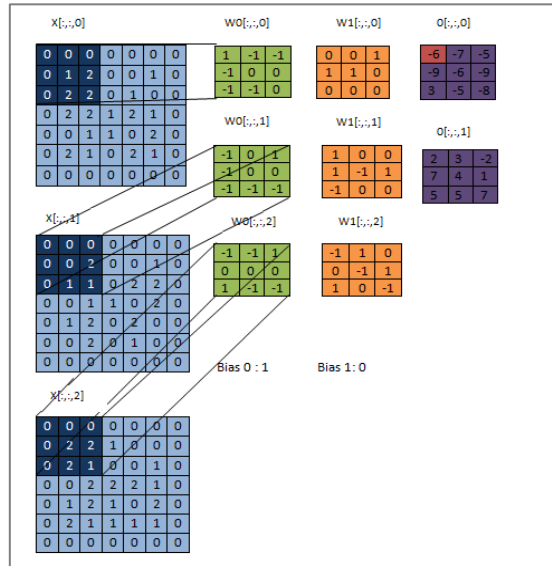
Evrimsel sinir ağları, derin öğrenme alanında çok sayıda uygulama için kullanılan bir sinir ağı mimarisi türüdür [40]. CNN yapısı özellikle görsel tanıma ve işleme görevlerinde etkilidir, ancak ses işleme, doğal dil işleme ve diğer alanlarda da kullanılabilirler. CNN’lerin temel özelliği girdi verilerini hiyerarşik şekilde işleyerek özelliklerin çıkarılmasına ve sınıflandırmasına yardımcı olmalarıdır.

CNN’ler, girdi verilerini evrim katmanları, havuzlama katmanları, tam bağlantılı katmanlar ve aktivasyon fonksiyonları gibi farklı katmanlardan oluşan bir mimari ile işler. Bu katmanlar, verilerin özelliklerini çıkarmak, boyutunu azaltmak ve sınıflandırmak için birlikte çalışır. Şekil 4.6’da CNN yapısının ağ mimarisi gösterilmiştir.



Şekil 4.6. CNN sinir ağı mimarisi [41]

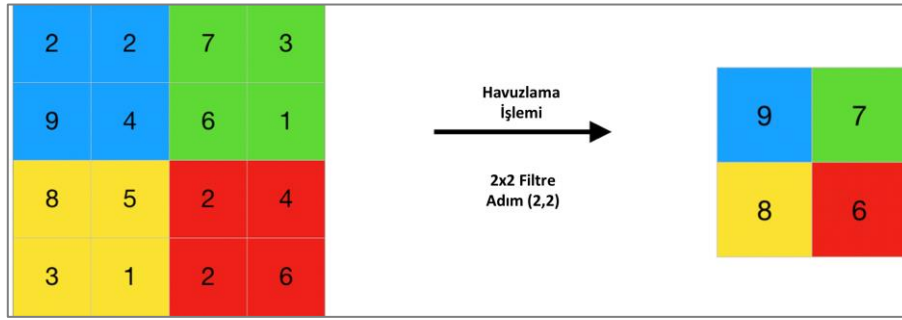
- **Evrişim Katmanları:** İlk katman olan evrişim katmanı, girdi verilerinin üzerinde küçük filtreler uygular. Bu filtreler, verilerin farklı yönlerini yakalamak için öğrenilir. Örneğin, bir görüntünün kenarlarını, renklerini veya dokularını tespit etmek için farklı filtreler kullanılabilir. Evrişim katmanları, filtrelerin uygulandığı her bölge için bir aktivasyon haritası üretir. Bu haritalar, verilerin özgün bir temsiliyi oluşturur.



Şekil 4.7. CNN’de evrişim işlemi [42]

Şekil 4.7’de görüldüğü gibi görüntü üzerine 3x3 matrisler ile filtre uygulaması yapılır. Piksel kaydırma işlemi ile 3x3 matrisler ile görüntü matrisi çarpılarak özellikleri çıkarılır.

- **Havuzlama Katmanı:** Evrişim katmanından sonra havuzlama katmanı gelmektedir. Bu katmanlar, aktivasyon haritalarının belirli bölgelerinde maksimum, minimum veya ortalama değerleri alarak daha küçük bir çıktı üretir. Bu işlem, hesaplama maliyetini düşürür ve aşırı uyumu önler. Havuzlama katmanları, verilerin ölçek ve dönüşüm değişikliklerine karşı dayanıklı olmasını sağlar. Genel olarak iki farklı havuzlama yöntemi mevcuttur. Bunlar maksimum havuzlama ve ortalama havuzlama yöntemleridir. Maksimum havuzlama yöntemi daha çok tercih edilmektedir. Maksimum havuzlama yöntemindeki amaç, yükseklik ve genişlik değerlerini azaltarak görüntü matrisinde baskın özellikler çıkarılması ve modelin etkili bir şekilde eğitilme sürecini sürdürmesidir.



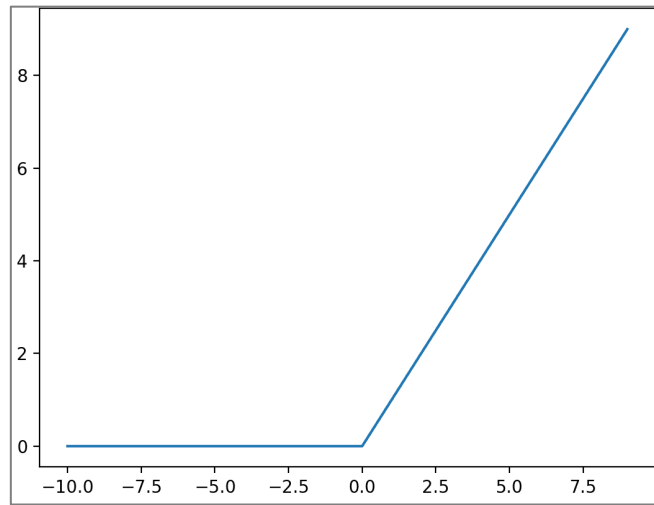
Şekil 4.8. Havuzlama işlemi

Şekil 4.8’de görüldüğü gibi görüntü matrisi 2x2’lik matrislere ayrılarak maksimum değerlerini çıkarılması sağlanmıştır.

- **Tam Bağlantılı Katmanlar:** Evrişimsel sinir ağlarının son katmanıdır. Tam bağlantılı katmanlar diğer katmanlarda oluşan özellik haritasını tek bir boyutlu vektöre dönüştürür. Bu vektör, sonraki katmanlara veya çıktı katmanına iletilir. Çıktı katmanı, verileri belirli sınıflara veya kategorilere atamak için kullanılır. Düzleştirilmiş vektör, çıkış katmanındaki nöronlara bağlanır. Bu nöronlar, giriş görüntüsünün hangi sınıfa ait olduğunu tahmin etmek veya bir regresyon görevinde sürekli bir değer tahmin etmek için kullanılır. En yüksek olasılığa sahip nöron, görüntünün sınıfı olarak seçilir.
- **Aktivasyon Fonksiyonları:** Evrişimsel sinir ağları her katman çıkışında çeşitli aktivasyon fonksiyonları kullanmaktadır. Aktivasyon fonksiyonlarının genel olarak görevi, her katmandaki nöronların çıkışını belirlemektir. CNN’de en fazla ReLU, sigmoid ve tanh aktivasyon fonksiyonları kullanılır.

ReLU, yani Rectified Linear Unit, girdi negatifse 0, pozitifse girdinin kendisi olarak tanımlanır. ReLu fonksiyonu, diğer aktivasyon fonksiyonlarına göre daha hızlıdır ve sadece pozitif değerlerle ilgilendiği için eşit dağılmış bir veri setinde yarı yarıya daha az işlem yapar. Ancak, 0 noktasında türevi hesaplanamaz. Bu fonksiyon, Vanishing Gradient Problemi'ni hafifletir. Vanishing Gradient Problemi, aktivasyon fonksiyonlarının verileri belirli bir aralığa sıkıştırması ve bu sıkıştırılan aralığın küçük olması hatta türevinin daha da küçük olmasıdır. Dolayısıyla belirli bir süre sonra dikkate alınmayacak kadar küçük optimizasyon parametrelerine ulaşması ve modelin tam olarak öğrenememesi sorundur. ReLu fonksiyonu, negatif değerler için türevi 0, pozitif değerler için türevi 1'dir. Eşitlik 4.1'de ReLu aktivasyon fonksiyon formülü gösterilmiştir. Şekil 4.9'da ise ReLu aktivasyon fonksiyonu grafiği gösterilmiştir.

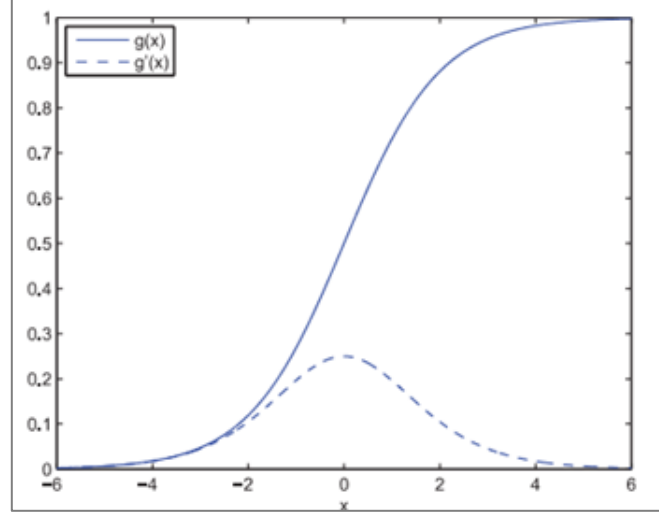
$$\sigma(x) = \begin{cases} \max(0, x), & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (4.1)$$



Şekil 4.9. ReLu aktivasyon fonksiyon grafiği

Sigmoid aktivasyon fonksiyonu, lojistik regresyon gibi ikili sınıflandırma problemlerinde kullanılmaktadır. Giriş değerini 0 ile 1 arasında bir değere dönüştürür ve bu sayede olasılıksal bir çıktı elde etmeyi sağlar. Eşitlik 4.2'deki fonksiyona giren x değeri ne kadar büyükse, fonksiyon o kadar 1'e yakın bir değer döndürmektedir. Fonksiyona giren x değeri 0 olduğunda 0,5 değerini döndürür. Şekil 4.10'da sigmoid aktivasyon fonksiyonu grafiği gösterilmiştir.

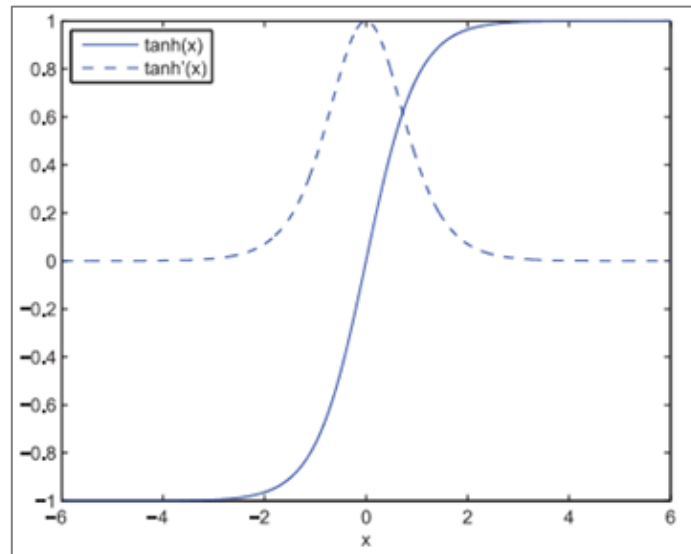
$$g(x) = \frac{1}{(1 + e^{-x})} \quad (4.2)$$



Şekil 4.10. Sigmoid aktivasyon fonksiyonu grafiği

Tanh, yani hiperbolik tanjant, yapay sinir ağlarında kullanılan bir aktivasyon fonksiyonudur. Tanh fonksiyonu, girdi değerlerini -1 ile 1 arasında bir değere dönüştürür. Sigmoid fonksiyonuna benzer, ancak daha geniş bir aralıkta değerler üretir. Şekil 4.11’de Tanh aktivasyon fonksiyon grafiği gösterilmiştir.

$$\tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (4.3)$$

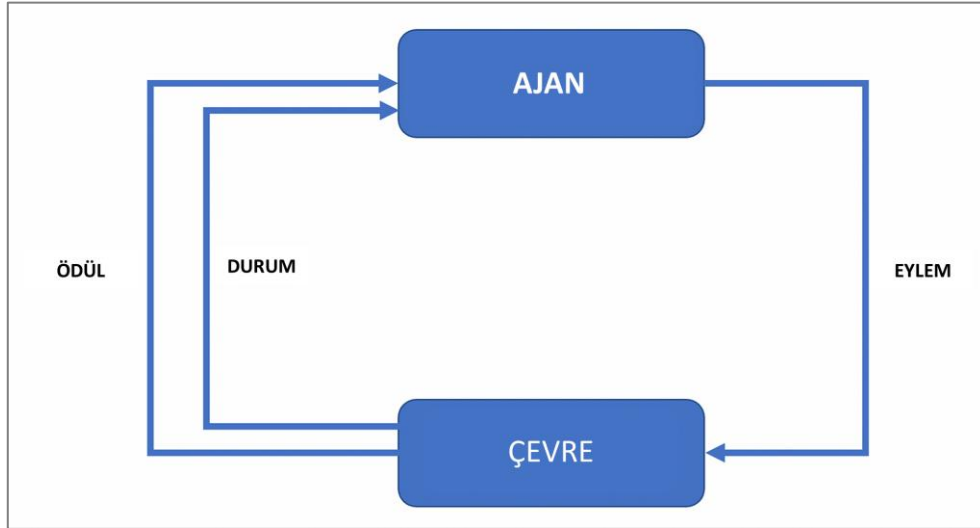


Şekil 4.11. Tanh aktivasyon fonksiyon grafiği

4.3.2. Yinelemeli sinir ağı

Yinelemeli sinir ağı (RNN), doğal dil işleme, İşitsel tanıma, zaman serisi analizi gibi sıralı verilerle çalışan bir yapay sinir ağı türüdür. RNN'ler önceki bilgileri hafızasında tutar ve bu bilgiler üzerine ekleme yaparak çalışır. Bu özellik sayesinde, zaman serileri, konuşma, metin ve benzeri düzgün bir sırası olan verilerin analizinde başarılıdır. Ancak uzun hafıza gerektiren durumlarda performansı düşmektedir.

RNN modelinde, bir ajan çevresindeki durumu gözlemleyerek eylemler gerçekleştirir ve bu eylemler sonucunda pozitif ya da negatif ödüller alır. Ajanın amacı, alınan toplam ödülünü maksimize etmektir. Şekil 4.12'de RNN modeli çalışma senaryosu gösterilmiştir.



Şekil 4.12. RNN modeli çalışma senaryosu

- Giriş katmanı: RNN'in ilk katmanıdır ve verileri ağı iletmek için kullanılır. Veri türüne bağlı olarak, metin, işitsel veya diğer veri formatlarından oluşabilir. Metin verileri genellikle kelime veya karakter düzeyinde kodlanırken, işitsel verileri genellikle spektrogramlar veya MFCC'ler gibi özellik vektörleri olarak temsil edilir.
- Yinelemeli Katmanlar: RNN'in en önemli bileşenidir. Yinelemeli hücrelerden oluşurlar ve ağın zaman bağımlılığını ve ardışık veri yapılarını modellemek için kullanılırlar. İki yaygın yinelemeli hücre tipi, Uzun Kısa Süreli Bellek (LSTM) ve Kapılı Rekürrent Birimler (GRU)'dir. Her yinelemeli hücre, mevcut girdiyi hesaplamak için önceki hücrenin çıktısını kullanır. Bu, ağın geçmiş bilgileri

hatırlamasını ve mevcut girdiye bağı olarak gelecekteki çıktıları tahmin etmesini sağlar.

- Çıktı Katmanı: RNN'in son katmanıdır ve genellikle bir sınıflandırma, regresyon veya dil oluşturma gibi bir görev için kullanılır. Sınıflandırma görevlerinde, çıktı katmanı genellikle sınıf olasılıklarını temsil eden bir aktivasyon fonksiyonu içerir. Regresyon görevlerinde ise çıktı katmanı doğrudan tahmin edilen değeri içerir.
- Kayıp Fonksiyonu: RNN'in eğitimi sırasında kullanılan bir hata metriği olarak işlev görür. Kayıp fonksiyonu, ağın tahminlerinin gerçek değerlerle ne kadar uyumlu olduğunu ölçer. Sınıflandırma görevlerinde yaygın olarak kullanılan kayıp fonksiyonları arasında kategori çapraz entropi ve ortalama kare hatası yer alır. Regresyon görevlerinde ise ortalama kare hatası veya ortalama mutlak hata gibi kayıp fonksiyonları kullanılabilir.
- Geri Yayılım Algoritması: RNN'in eğitimi sırasında kullanılan bir optimizasyon algoritmasıdır. Geri yayılım algoritması, kaybı azaltmak için ağı geriye doğru geçerek ağırlıkları ve biasları günceller. Bu, ağırlıkların ve biasların gradyan iniş yöntemiyle ayarlanmasını sağlar, böylece ağın tahminleri gerçek değerlere daha yakın hale gelir.

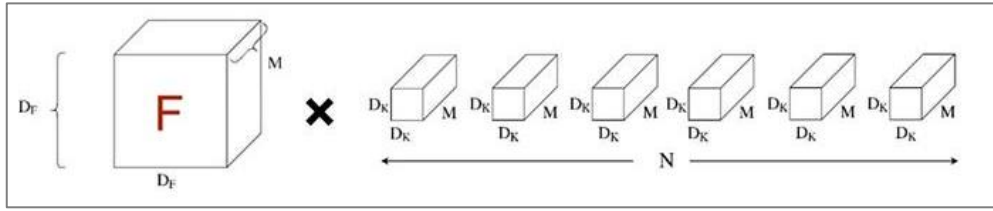
4.3.3. MobileNets

MobileNets, özellikle mobil ve gömülü cihazlar için tasarlanmış ve verimli bir konvolüsyonel sinir ağı mimarisidir. En önemli özellikleri mobil cihazlar gibi kaynak sınırlı ortamlarda kullanım için optimize edilmeleridir [43]. MobileNets, yüksek doğruluk seviyelerini korurken, daha az hesaplama gücü ve bellek gereksinimiyle çalışabilen derin öğrenme uygulamaları için idealdir. Bu mimari, Google araştırmacıları tarafından geliştirilmiş ve açık kaynak olarak sunulmuştur.

- MobileNetV1: MobileNets serisinin ilk sürümüdür ve Google araştırmacıları tarafından 2017 yılında tanıtılmıştır [44]. MobileNet V1'in temel fikri, standart konvolüsyon işlemini iki ayrı işleme ayırmaktır: derinlikli konvolüsyon ve noktasal konvolüsyon. Derinlikli konvolüsyon, girdi kanallarını tek tek filtreler, böylece her kanal için bir özellik haritası oluşturur. Noktasal konvolüsyon, tüm kanalları

birleştirerek çıkış kanallarını oluşturur. Bu şekilde hem konvolüsyon filtrelerinin sayısı hem de çarpma-toplama işlemlerinin sayısı azaltılır.

MobileNets'in en önemli özelliklerinden biri derinlik ayrıştırma olarak adlandırılan bir konvolüsyon tekniğidir. Her giriş kanalına tek bir filtre uygulamak için kullanılır. Bu, filtrelerin tüm giriş kanallarına uygulandığı standart evrişimden farklıdır. Şekil 4.13'de standart bir konvolüsyon modeli gösterilmektedir.

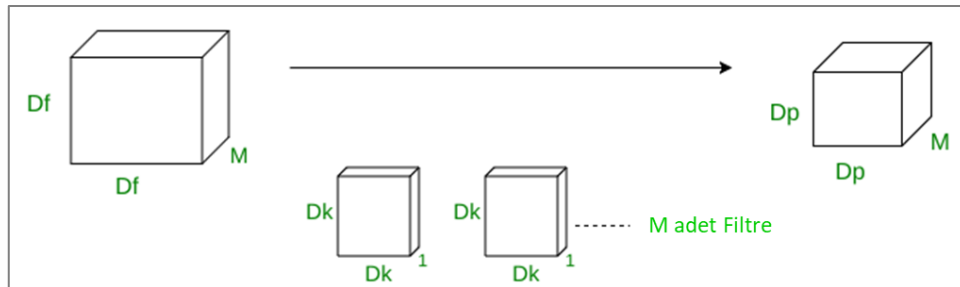


Şekil 4.13. Standart konvolüsyon işlemi [45]

Hesaplama maliyeti, bir CNN modelinin eğitimi ve kullanımı sırasında gereken zaman ve kaynakların miktarını belirlemeye yardımcı olan önemli bir metriktir. Standart konvolüsyondaki hesaplama maliyeti Eşitlik 4.4'de gösterilmiştir.

$$\text{Hesaplama Maliyeti (Standart)} = Dk * Dk * M * N * Df * Df \quad (4.4)$$

Hesaplama maliyetindeki Dk Kernel derinliği anlamına gelir. Df öznetelik derinliği, M çıktı filtre sayısı ve N örnek sayısı anlamına gelmektedir. Standart bir evrişim için hesaplama maliyeti, giriş ve çıkış kanallarının sayısına ve giriş özellik haritasının ve evrişim çekirdeğinin uzamsal boyutlarına çarpımsal olarak bağlıdır. Şekil 4.14'de derin ayrıştırma konvolüsyon modeli gösterilmektedir.

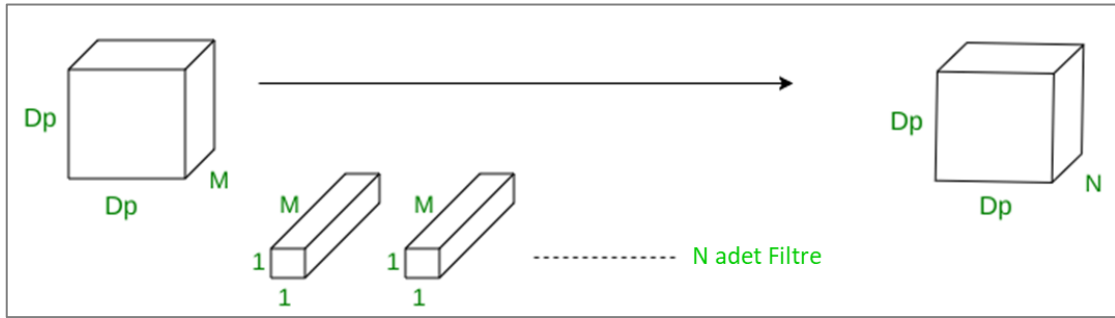


Şekil 4.14. Derin ayrıştırma konvolüsyon işlemi [46]

$$\text{Hesaplama Maliyeti (Derin Ayırıştırma)} = Dk * Dk * M * Df * Df \quad (4.5)$$

Derin ayırıştırma konvolüsyonda, her giriş kanalına tek bir filtre uygulanmaktadır. Standart evrişimden farklı olarak, filtreler tüm giriş kanallarına uygulanmaz. Bu sayede, modelin parametre sayısı ve hesaplama gücü önemli ölçüde azaltılır. Böylece standart evrişime göre hesaplama maliyeti de azalmaktadır.

Noktasal evrişimde derin ayırıştırma gibi MobileNets modelinde önemli bir yer tutmaktadır. Noktasal evrişim, genellikle derin ayırıştırma konvolüsyonlarının bir parçası olarak kullanılır. Bu yöntem bir 1x1 konvolüsyonu ifade eder ve çok kanallı girdi verisindeki kanallar arasındaki etkileşimi modellemek için kullanılır. Noktasal evrişim, derin ayırıştırma konvolüsyonun çıktılarını birleştirmek için kullanılan ek bir katmandır. Şekil 4.15’de noktasal evrişim işlemi gösterilmiştir.



Şekil 4.15. Noktasal evrişim işlemi

Noktasal evrişimin hesaplama maliyeti Eşitlik 4.6’da gösterilmiştir.

$$\text{Hesaplama Maliyeti (Noktasal)} = M * N * Df * Df \quad (4.6)$$

Dolayısıyla, derinlemesine ayrılabilir konvolüsyonların toplam hesaplama maliyeti Eşitlik 4.7’deki gibi olmalıdır.

$$(Dk * Dk * M * Df * Df) + (M * N * Df * Df) \quad (4.7)$$

Bu formülü, standart konvolüsyonun hesaplama maliyeti ile karşılaştırıldığında Eşitlik 4.8’deki gibi bir azalma elde edilecektir.

$$\text{Maliyetteki Azalma} = \frac{(Dk * Dk * M * Df * Df) + (M * N * Df * Df)}{(Dk * Dk * M * N * Df * Df)} = \frac{1}{N} + \frac{1}{D_k^2} \quad (4.8)$$

Sonuç olarak derinlemesine ayrılabilir konvolüsyon ağının, stardart bir konvolüsyon ağına göre 100 kat daha az çarpma işlemi gerçekleştirdiği anlamına gelir.

- MobileNetV2: Google tarafından MobileNet serisinin ikinci versiyonu olarak geliştirilmiş hafif bir derin sinir ağı mimarisidir. MobileNet V2'nin en önemli yeniliklerinden biri, ters çevrilmiş kalıntı bloklar ve bottleneck mimarisi adı verilen iki mimari özelliğin uygulanmasıdır. Bu yapılar sayesinde MobileNetV2, önceki sürüme göre daha az enerji harcayarak daha yüksek doğruluk sağlamaktadır.

Ters çevrilmiş kalıntı blokları, bir çeşit tersine eğitim yöntemiyle ağı daha fazla çeşitlilik ve verimlilik elde etmesini sağlamak için kullanılan bir mimariyi ifade eder. Bu bloklar, geleneksel kalıntı bloklarının tersine, ağı içerisindeki daha düşük katmanların önceki katmanlardan gelen daha yüksek katmanlara bağlanmasını sağlar. Geleneksel bir kalıntı bloğunda, önceki katmanın çıkışı (identity mapping), bir sonraki katmana (genellikle daha yüksek seviyeli bir katmana) eklenir. Ters çevrilmiş kalıntı bloklarında ise, önceki katmanın çıkışı, daha düşük seviyeli bir katmana eklenir. Bu, daha yüksek seviyeli katmanların daha düşük seviyeli katmanların çıktılarını kullanmasıyla gerçekleşir ve ağı daha fazla özgüllük öğrenmesine yardımcı olur. Ters çevrilmiş bir kalıntı bloğu, iki konvolüsyon katmanından oluşur: bir 1x1 boyutlu konvolüsyon katmanı ve bir 3x3 boyutlu konvolüsyon katmanıdır. 1x1 konvolüsyon katmanı, kanalların boyutunu değiştirirken, 3x3 boyutlu konvolüsyon katmanı özellik haritalarını çıkarır. İki katman arasında, bir kalıntı bağlantısı bulunur. Bu bağlantı, girdiyi doğrudan bloğun çıkışına ekler.

Bottleneck Mimarisi, ağı daha hafif hale getirmek için kullanılan bir tekniktir. Bu mimari, daha büyük ve daha hesaplama yoğunluğuna sahip bir modelin performansını korurken, daha küçük bir model boyutu elde etmek için kullanılır. Daralan aşama, genişleme aşaması ve projeksiyon aşaması olmak üzere 3 aşamadan oluşmaktadır.

- ✓ Daralan aşamada, giriş verisi boyutunu azaltmak için 1x1 konvolüsyon katmanları kullanılır. Bu, veri boyutunu azaltır ve modelin hesaplama maliyetini düşürür.

- ✓ Genişleme aşamasında, daha büyük boyutlu 3x3 konvolüsyon katmanları kullanılarak verinin özelliklerini genişletmek için işlem yapılır. Bu aşama, daha fazla özellik öğrenilmesine ve modelin özgünlüğünün artmasına yardımcı olur.
- ✓ Projeksiyon aşamasında, çıktı boyutunu tekrar azaltmak için bir başka 1x1 konvolüsyon katmanı kullanılır. Bu adım, modelin çıktı boyutunu uygun bir şekilde ayarlar.
- MobileNetV3: Mobil cihazlarda yüksek performanslı görsel algılama görevlerini gerçekleştirmek için tasarlanmıştır ve önceki MobileNet sürümlerine göre daha verimli ve etkili bir mimari sunmaktadır. Bu, mobil cihazlarda derin öğrenme uygulamalarının geliştirilmesi ve dağıtılması için önemli bir adımdır. MobileNet V3'ün temel özellikleri arasında, verimli mobil yapı taşları, blok bazında yapılan sinir mimarisi araması (NAS) ve katman bazında yapılan NetAdapt algoritması bulunur. Bu yeni sürüm, önceki sürümlere göre birçok iyileştirme sunar. MobileNets v3'te sıkıştırma ve uyarılma katmanları eklenmiştir. Bu katmanlar, farklı kanallara girdiye göre değişken ağırlıklar vererek çıktı özellik haritalarının oluşturulmasına yardımcı olur.

MobileNet V3, h-swish adı verilen yeni bir aktivasyon fonksiyonu kullanır. H-swish aktivasyon fonksiyonu, girdiler ile sigmoid fonksiyonunun çarpımını çıktı olarak üretmektedir. H-swish aktivasyon fonksiyonunun formülü Eşitlik 4.9'da gösterilmektedir.

$$H - swish(x) = x * sigmoid(x) \quad (4.9)$$

H-swish, Doğrusal olmayan bir fonksiyondur, bu da karmaşık problemlerin çözümünde daha etkili olabileceği anlamına gelir. Türevlenebilir bir fonksiyondur, bu da geri yayılım algoritmasının uygulanmasını kolaylaştırır.

4.4. TinyML Araç Setleri

TinyML araç setleri, küçük cihazlarda makine öğrenimi uygulamalarını geliştirmek ve çalıştırmak için kullanılan yazılım ve donanım bileşenleridir. TinyML araç setleri, düşük güç tüketimi, sınırlı bellek ve işlem kapasitesi gibi zorlukları aşmak için tasarlanmıştır.

4.4.1. Donanım

TinyML, gömülü sistemlerde, özellikle düşük güç tüketimi olanlar üzerinde çalışır. TinyML destekli donanımlar aşağıdaki özellikleri kapsamaktadır.

- **Küçük Boyutlu:** TinyML destekli donanımlar, genellikle birkaç santimetre veya daha küçük boyutlardadır. Bu sayede, giyilebilir cihazlar, akıllı oyuncaklar, ev aletleri veya endüstriyel ekipmanlar gibi farklı uygulama alanlarına entegre edilebilirler.
- **Düşük Güç Tüketimi:** TinyML destekli donanımlar, enerji verimliliği yüksek olan mikrodenetleyiciler ve algoritmalar kullanır. Bu sayede, batarya ömrü uzatılır ve sık sık şarj etme ihtiyacı azaltılır. Bazı TinyML destekli donanımlar, güneş enerjisi, termoelektrik veya kinetik enerji gibi yenilenebilir kaynaklardan da beslenebilirler.
- **Yerel İşlem Yeteneği:** TinyML destekli donanımlar, verileri buluta göndermeden yerel olarak işleyebilirler. Bu sayede, gecikme süresi kısaltılır, bant genişliği tasarrufu sağlanır ve gizlilik korunur. Bazı TinyML destekli donanımlar, bulut ile iletişim kurmadan da bağımsız olarak çalışabilirler.

Tablo 4.1’de TinyML destekli ve bu çalışmada model eğitimi için kullanılan edge impulse tarafından desteklenen donanımlar gösterilmiştir.

Tablo 4.1. TinyML destekli donanımlar

Donanım	İşlemci	İşlemci Hızı	Bellek (RAM)	Bellek (ROM)	Fiyat (\$)
Arduino Nano 33 BLE Sense	Cortex M4F	64 MHz	256 KB	1024 KB	40.50
Arduino Nicla Vision	Cortex-M7	480 MHz	128 KB	1024 KB	148.14
Arduino Portenta H7	Cortex-M7	480 MHz	512 KB	2048 KB	99
Akida PCIe Board	Cortex M4	300 MHz	524288 KB	16384 KB	500
ESP32-CAM	ESP32	240 MHz	4096 KB	4096 KB	30
nRF52840 DK	Cortex-M4F	64 MHz	256 KB	1024 KB	95
Raspberry Pi RP2040	Cortex-M0	133 MHz	264 KB	2048 KB	6

4.4.2. Yazılım

TinyML ile kullanılan yazılımlar, makine öğrenimi modellerini güç, işlem ve bellek kısıtlamaları olan mikro denetleyicilere yerleştirmek için tasarlanmış olan yazılımlardır.

- TensorFlow Lite Micro: Google'ın açık kaynaklı makine öğrenimi çerçevesi olan TensorFlow'un bir sürümüdür. TensorFlow Lite Micro, makine öğrenimi modellerini eğitmek ve mikrodenetleyicilere dağıtmak için optimize edilmiştir.
- Edge Impulse: TinyML uygulamalarının geliştirilmesine odaklanan bir platformdur. Edge impulse, içerisinde veri toplama, işleme ve dağıtma oluşturan geniş kapsamlı bir yapay zekâ platformudur. En önemli özelliği, bulut hizmetlerini kullanmasıdır. Edge impulse, kullanıcıların sensör verisi toplamasına, etiketlemesine ve bu sensör verilerini kullanarak makine öğrenmesi modellerini eğitmesine yardımcı olur. Ayrıca bu modellerin gömülü cihazlara dağıtılmasını sağlar. Bu sayede makine öğrenmesi modelleri, çevrimdışı durumlar dahil olmak üzere daha fazla esneklik ve hızla çalışabilir.
- Google Coral: Yerel makine öğrenmesi çözümleri için Google'ın geliştirdiği bir platformdur. Coral'in temel amacı yapay zekâ ve makine öğrenmesi modellerinin bulut yerine yerel cihazlarda daha hızlı ve verimli şekilde çalışmasını sağlamaktır. Google Coral, genellikle görüntü sınıflandırma, nesne algılama, İşitsel işleme, konuşma tanıma ve benzeri uygulamalar için kullanılır. Bu platform, TensorFlow Lite ile entegre olarak çalışır ve ML modellerini cihaz üzerinde hızlı ve etkin bir şekilde çalıştırmak için kullanılır.
- Amazon SageMarker: Amazon SageMarker, Amazon Web Services'in (AWS) sunduğu bir makine öğrenimi platformudur. Amazon SageMarker, ML geliştiricilerine ve veri bilimcilerine, veri hazırlama, model eğitme, model dağıtma ve model izleme gibi ML yaşam döngüsünün tüm aşamalarını kolaylaştıran araçlar ve hizmetler sağlar.
- uTensor: Gömülü sistemler ve mikrodenetleyiciler için bir makine öğrenmesi çerçevesidir. Bu framework, TensorFlow modellerini küçük, düşük güçlü cihazlarda (genelde IoT cihazları) çalıştırmak için kullanılır. Bu, veri merkezlerine veya buluta bağlı olmayan durumlarda çok değerli olabilir. uTensor, bir TensorFlow modelini alır ve onu yürütmek için uygun C++ koduna çevirir. Oluşturulan kod daha sonra belirli bir gömülü cihaza (genellikle bir ARM Cortex-M mikrodenetleyici) yüklenir.

- NanoEdge AI Studio: Nesnelerin interneti cihazlarına yerel ve düşük güç tüketimli AI yetenekleri kazandırmayı sağlayan bir yazılım çözümüdür. Yazılımın özellikleri arasında AI model eğitim ve oluşturma, model optimizasyonu, AI algoritmalarının entegrasyonu ve gözden geçirilmiş CPU kullanımı yer almaktadır.

TinyML için en çok kullanılan 2 platform edge impulse ve tensorflow lite'dır. Şekil 4.16'da bu 2 platform karşılaştırılmıştır.

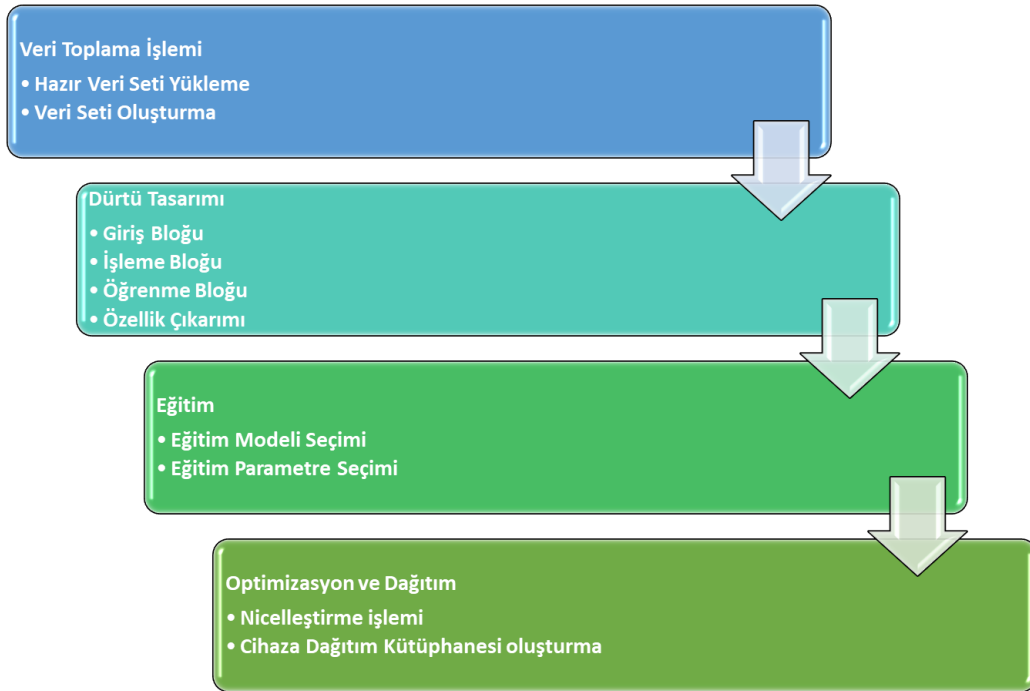
Edge Impulse	TensorFlow Lite
Kodlama bilgisi istemez. Kullanıcı dostu arayüze sahiptir. Optimizasyon otomatik olarak yapılır.	Makine öğrenimi ve kodlama bilgisi ister. Model optimizasyonu için daha derin bir bilgi gerekir.
Ücretsiz seçenekleri vardır ancak daha gelişmiş özellikler için ücret planları mevcuttur.	Ücretsiz ve açık kaynaklıdır.
Veri toplama, etiketleme ve veri kümesinin işlenmesini sağlar.	Genellikle geliştiricilerin önceden etiketlenmiş veri kümesi üzerinde çalışır.
Veri toplamadan başlayarak eğitim ve dağıtım işlemlerini yapmaktadır.	Model eğitimi olmaz. Eğitilmiş model optimize edilir.

Şekil 4.16. Edge Impulse ile TensorFlow Lite kıyaslaması

5. TinyML İLE GÖRSEL İŞİTSEL ANAHTAR KELİME TESPİTİ

Son yıllarda yapılan AKT çalışmaları genel olarak işitsel tabanlı yapılmaktadır. Gürültüsüz bir ortamda yapılan anahtar kelime tespit çalışmaları çok başarılı sonuçlar vermektedir. Ancak gürültülü ortamlarda, özellikle sanayi ve açık alanlarda yapılan çalışmalar başarıyı olumsuz yönde etkilemektedir. Görsel işitsel anahtar kelime tespiti hem görüntü hem de işitsel verilerini birlikte analiz ederek, sadece işitsel verilerine dayanan geleneksel anahtar kelime tespitinden daha yüksek doğruluk ve güvenilirlik sağlayabilir.

Bu bölümde ilk olarak edge impulse platformunda ayrı ayrı görsel ve işitsel modellerin oluşturulması ve cihaza dağıtılması anlatılmıştır. Daha sonrasında ise modellerin doğrusal ağırlıklı birleştirme metodu ile birleştirilmesi anlatılmıştır. Şekil 5.1’de model oluşturma adımları gösterilmiştir.



Şekil 5.1. Edge Impulse platformunda model oluşturma adımları

5.1. Görsel Anahtar Kelime Tespiti

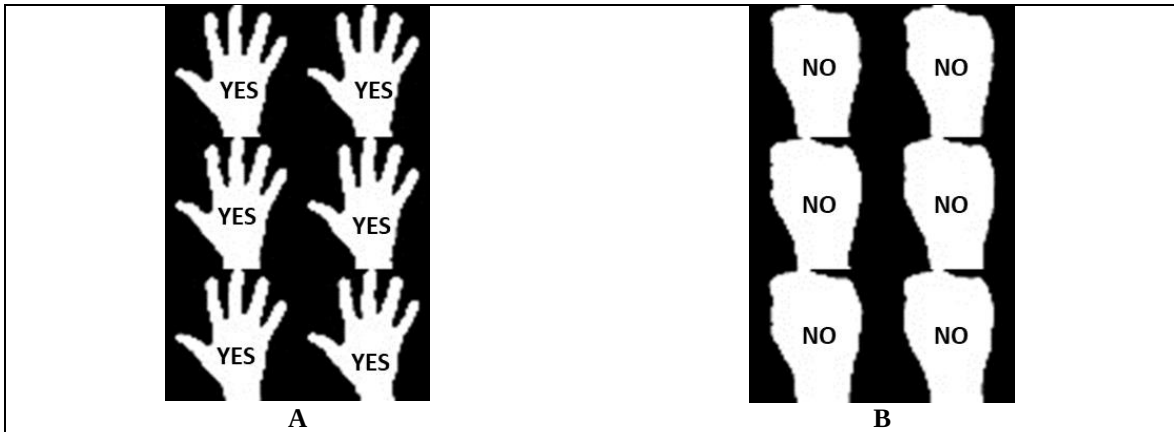
TinyML uygulamalarında görsel AKT, bir görüntüdeki nesnelerin veya özelliklerin otomatik olarak tanınmasını ve sınıflandırılmasını sağlayan bir süreçtir. Algoritmaların ve

modelleme tekniklerinin resim ve videolardaki nesneleri, desenleri ve özellikleri tanıma ve bunları önceden tanımlanmış sınıflara atama yeteneğini ifade eder.

Bu çalışmada görsel veri sınıflandırma işlemini ESP32-CAM modülü kullanılmıştır. ESP32-CAM, bir ESP32 WiFi modülü içeren ve dahili bir video kamera barındıran bir geliştirme kartıdır. ESP32-CAM, entegre bir mikro denetleyiciye sahip bir modüldür ve WiFi + Bluetooth bağlantısına ek olarak, entegre bir video kamera ve depolama için bir microSD yuvası içerir.

5.1.1. Görsel AKT’de veri toplama

Klasik bir makine öğreniminde olduğu gibi TinyML ile de bir model oluştururken ilk yapılan işlem modelin veri setini hazırlamaktır. Bu çalışmada “Hand Gesture Recognition Dataset” kullanılmıştır [47]. Bu veri setinde 20 farklı el hareketinden 24000 fotoğraf mevcuttur. Her sınıf için 900 eğitim verisi 300 test verisi bulunmaktadır. Veriler %75 eğitim, %25 test verileri olacak şekilde ayrılmıştır. Veri seti hazırlanırken 1 kişinin farklı konumlardaki el hareketleri kullanılmıştır. Bu çalışmada, 2 farklı el durumu ile tespit yapılmıştır. Şekil 5.2’de çalışmada kullanılan el durumları gösterilmiştir.



Şekil 5.2. Açık el hareketi (A) ve kapalı el hareketi (B)

5.1.2. Görsel AKT’de dürtü tasarımı

TinyML uygulamalarında veri seti oluşturulduktan sonraki işlem dürtü tasarımı yapmaktır. Edge Impulse platformunda dürtü tasarımı işlemi, veri setinden alınan verileri işleyip özellikler çıkarmak ve yeni verileri sınıflandırmak için bir makine öğrenimi akışı

oluşturmaya yaramaktadır. Bir dürtü 3 ana yapıdan meydana gelmektedir. Bunlar giriş bloğu, işleme bloğu ve öğrenme bloğudur.

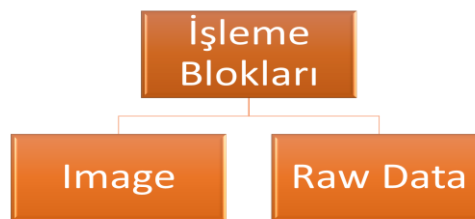
Görsel veriler için boyutlandırma modu bu blokta seçilmektedir. Bu çalışma için “Squash” boyutlandırma modu seçilmiştir. Şekil 5.3’de veri boyutlandırma modları gösterilmiştir.



Şekil 5.3. Edge Impulse platformundaki veri boyutlandırma modları

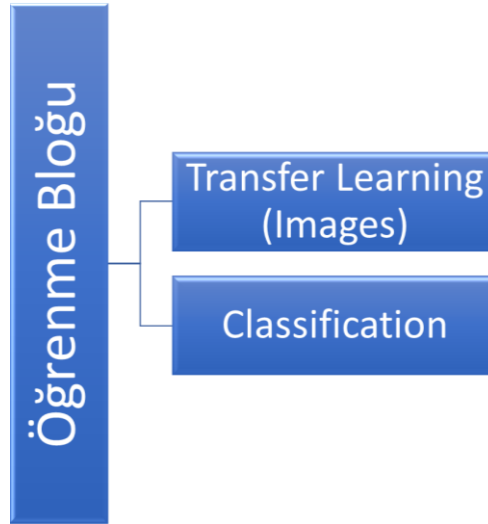
İşleme bloğu, girdi verilerini işlemek ve modelin daha iyi performans göstermesini sağlamak için kullanılan bir yapıdır. Özellikle görüntü sınıflandırma gibi görevlerde işleme bloğu, görüntüleri ön işlemek, özellik çıkarımı yapmak ve modelinize uygun bir formata dönüştürmek için kullanılır.

DSP (Digital Signal Processing), işleme bloğunda kullanılır. DSP, analog sinyalleri dijital verilere dönüştürmek ve bu verileri işlemek için kullanılan bir teknolojidir. DSP, işitsel, görüntü, video, radar, biyomedikal ve iletişim sistemleri gibi birçok alanda uygulama bulmaktadır. DSP, analog sinyalleri örnekleme, niceleme ve kodlama adımlarıyla dijital sinyallere çevirir. Dijital sinyaller, sayısal işlemciler, bilgisayarlar veya özel donanımlar tarafından analiz edilebilir, filtrelenebilir, modifiye edilebilir veya sentezlenebilir. DSP'nin avantajları arasında yüksek hız, düşük maliyet, yüksek doğruluk, esneklik ve geliştirilebilirlik sayılabilir. DSP'nin dezavantajları ise analog sinyallerin kaybı, örnekleme hızının sınırlı olması ve işlem karmaşıklığıdır. Şekil 5.4’de edge impulse tarafından kullanılan işleme blokları verilmiştir.



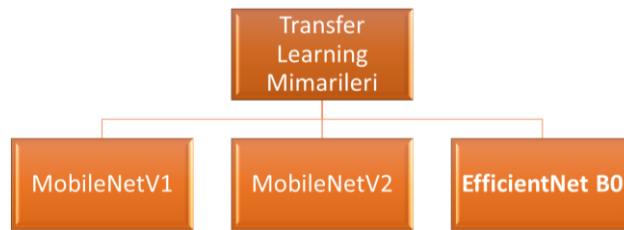
Şekil 5.4. Görüntü sınıflandırmada kullanılan işleme blokları

Image işleme bloğu veri setinin resim uzantısı olduğu durumlarda kullanılmaktadır. Raw data ise veri setinin sayısal olması gerekmektedir. Öğrenme bloğu, verileri eğitmek ve makine öğrenimi modelleri oluşturmak için kullanılan bir araçtır. Bu, bir makine öğrenimi modelinin eğitimini veya tahmin yapmasını gerçekleştiren bir bileşendir. Şekil 5.5’de edge impulse tarafından desteklenen öğrenme blokları mevcuttur.



Şekil 5.5. Görüntü sınıflandırmada kullanılan öğrenme blokları

Şekil 5.5’de gösterilen öğrenme blokları edge impulse’da çeşitli yapay sinir ağı mimarileri ile kullanılmaktadır. Şekil 5.6’da Transfer learning mimarileri ve Şekil 5.7’de Classification mimarileri gösterilmektedir.



Şekil 5.6. Transfer Learning mimarileri

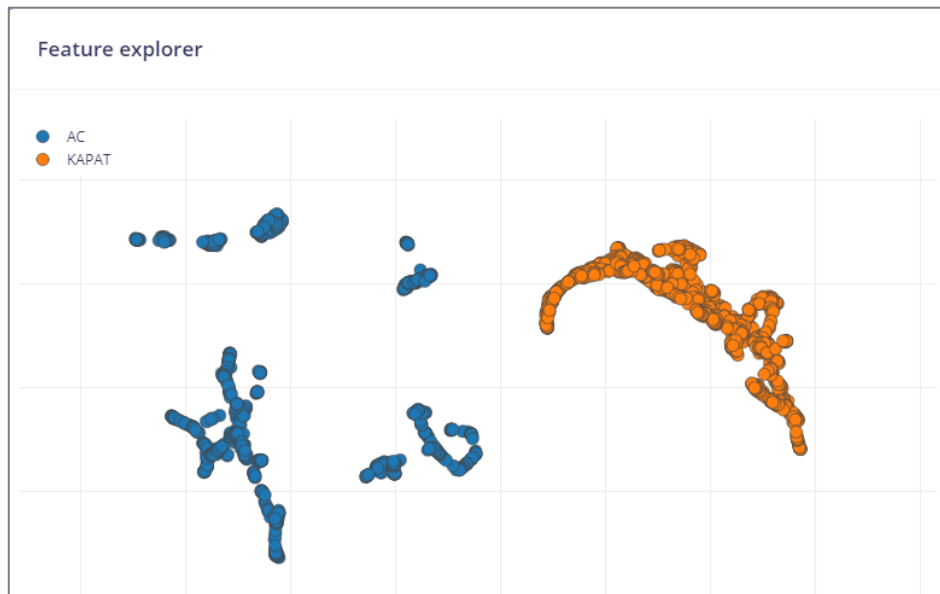
EfficientNet, Google Brain tarafından geliştirilen ve hem hesaplama verimliliği hem de performans açısından optimize edilmiş bir derin öğrenme modeli ailesidir. EfficientNet B0, Convolutional Neural Network tabanlı bir mimari kullanır ve öğrenilebilir parametrelerle birlikte derin bir modeldir.



Şekil 5.7. Classification mimarileri

Dürtü tasarımının son adımında ise verinin özellikleri çıkarılır. Özellik çıkarımı, verilerden özellikler oluşturmaya yarar. Bu, makine öğrenim modellerini eğitmek ve test etmek için kullanılacak veri girişlerinin hazırlanmasına yardımcı olur. Bu işlem, verileri daha anlamlı bir formata dönüştürmek ve modelin daha iyi sonuçlar elde etmesine yardımcı olmak için kullanılır.

Edge impulse’da özellik çıkarım işlemi seçilen işleme bloğuna göre yapılmaktadır. Görüntü verileri işlenmeden önce renk derinliği ayarlanır. Gri skala veya renkli şekilde olmak üzere 2 seçenek bulunmaktadır. Bu çalışmada modelin fazla boyut kaplamaması için gri skala seçilmiştir. Dürtü tasarımı sonucunda bir özellik gezgini vermektedir. Özellik gezgini verilerin kalitesini gösterir. Birbirine yakın gruplanmış verilerin özelliklerinin iyi çıkarıldığı anlamına gelmektedir. Şekil 5.8’de Hand Gesture Dataset için özellik gezgini görüntülenmektedir.



Şekil 5.8. Görsel sınıflandırma sonucu elde edilen özellik gezgini

5.1.3. Görsel AKT’de eğitim

Eğitim işlemi Edge Impulse üzerinde oluşan yapay sinir ağı modelinin yapılandırılmasını sağlar. Sinir ağı modeline kullanılmak istenen katmanların, hiper parametrelerin ve diğer ayarların yapılandırılmasına olanak tanır. Edge Impulse platformundan alınan parametreler Tablo 5.1’de gösterilmiştir.

Tablo 5.1. Görsel AKT’de kullanılan eğitim parametreleri

Parametre	Miktarı / Durumu
Eğitim Döngüsü	20
Öğrenme Oranı	0.0005
Veri Artırımı	Yok
Dropout	0.1

- **Eğitim Döngüsü (Training Cycles):** Bir modelin eğitim verileri üzerinde iteratif olarak dönerek güncellendiği adımları ifade eder. Bu adımlar, modelin verileri işleyerek içerisindeki parametreleri (ağırlıklar ve bias'lar gibi) ayarlamayı ve eniyilemeyi hedefler. Edge Impulse tarafından varsayılan değer 20’dir.
- **Öğrenme Oranı (Learning Rate):** Öğrenme oranı, modelin ağırlıklarını güncellemek için kullanılan adım boyutunu belirler. Nöron parametrelerinin güncellenmesi için kullanılan bir hiper parametredir. Geriye doğru türev alınarak hesaplanan nöron parametrelerinin güncellenmesi, öğrenme oranı ile çarpılarak yapılır. Edge Impulse tarafından varsayılan değer 0.0005’dir.
- **Veri Artırımı (Data Augmentation):** Veri artırımı, eğitim veri kümesinin boyutunu yapay olarak genişletmek için kullanılan bir tekniktir. Bu teknik, görüntülerin değiştirilmiş versiyonlarını oluşturarak veri kümesindeki görüntü sayısını artırır. Bu sayede, modelin daha fazla veri üzerinde eğitilmesi sağlanır ve daha yetenekli modeller elde edilir. Edge Impulse varsayılan olarak bu özelliği kullanmamaktadır.
- **Dropout:** Dropout parametresi, aşırı öğrenmeyi önlemek için kullanılan bir tekniktir. Dropout, eğitim sırasında rastgele seçilen nöronların çıkartılması işlemidir. Bu sayede, ağırlık farklı alt kümeleri üzerinde eğitilmesi sağlanır ve aşırı öğrenme riski azaltılır. Edge Impulse, bu parametreyi varsayılan olarak 0.1 seçmiştir.

Edge impulse platformu kullanılarak görsel sınıflandırma tamamlanmıştır. Şekil 5.9’da eğitim sonucunu göstermektedir. Eğitim sonucunda edge impulse platformunda bir karışıklık matrisi verilmektedir.

		GERÇEK	
TAHMİN		AÇIK EL	KAPALI EL
	AÇIK EL	100%	0%
	KAPALI EL	0%	100%
	F1 SKORU	1.00	0

Şekil 5.9. Görsel AKT eğitimi sonrası Edge Impulse’da hesaplanmış karışıklık matrisi

F1 skoru, bir modelinin performansını ölçen bir metriktir. Hassasiyet ve geri çağırma parametrelerinin harmonik ortalaması olarak hesaplanır. Eşitlik 5.1 hassasiyet hesaplamasını göstermektedir. Hassasiyet, modelin pozitif olarak tahmin ettiği örneklerin gerçekten pozitif olma oranını ifade etmektedir.

$$Hassasiyet = \frac{TP}{TP + FP} \quad (5.1)$$

Geri çağırma, gerçek pozitif örneklerin ne kadarının doğru bir şekilde tanındığını ifade etmektedir. Eşitlik 5.2 geri çağırma hesaplamasını göstermektedir.

$$Geri \text{ Çağırma} = \frac{TP}{TP + FN} \quad (5.2)$$

Eşitlik 5.1 ve Eşitlik 5.2’deki TP, gerçek pozitif örneklerin doğru bir şekilde pozitif olarak tahmin edilmesi anlamına gelmektedir. FP, gerçek negatif örneklerin yanlış bir şekilde pozitif olarak tahmin edilmesini göstermektedir. FN, gerçek pozitif örneklerin yanlış bir şekilde negatif olarak tahmin edilmesini göstermektedir. F1 skoru, Eşitlik 5.3’de gösterilmiştir.

$$F1 = 2 \times \frac{Hassasiyet \times Geri \text{ Çağırma}}{Hassasiyet + Geri \text{ Çağırma}} \quad (5.3)$$

F1 skoru, 0 ile 1 arasında bir değer alır. 1, en iyi performansı temsil ederken, 0 en kötü performansı temsil eder.

5.1.4. Görsel AKT’de dağıtım ve niceleme

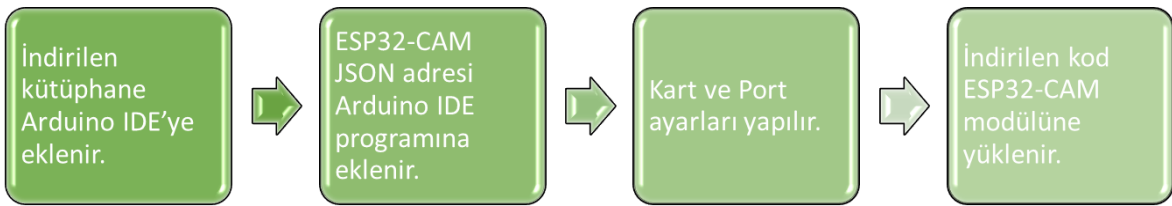
Edge Impulse’da dağıtım işlemi, geliştirilen yapay zekâ ve makine öğrenmesi modellerinin gerçek cihazlara uygulanabilir hale getirilmesi sürecidir. Dağıtım, modelin cihazlarda yerel olarak çalıştırmasını sağlamak için gerekli olan adımları içerir.

Edge Impulse, makine öğrenim modellerinin boyutunu ve hesaplama maliyetini azaltmak için niceleme tekniğini kullanır. Tablo 5.2’de niceleme tekniği ile optimize edilmiş ve edilmemiş modelin farkları gösterilmektedir.

Tablo 5.2. Görsel AKT’de optimizasyon sonucu Edge Impulse tarafından hesaplanan değerler

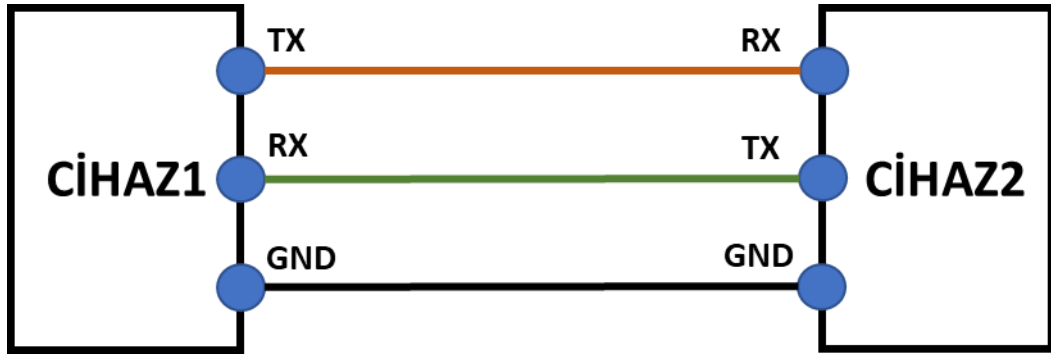
TEKNİK	RAM (kBYTE)	FLASH RAM (kBYTE)	GECİKME (MS)
Niceleme uygulanmış model	170.4	584,5	809
Niceleme uygulanmamış model	236,7	1600	1704

ESP32-CAM Arduino IDE programı üzerinden programlanabilmektedir. Edge impulse ESP32-CAM modülünü Arduino IDE programında direkt olarak çalıştırmayı sağlayan kütüphaneyi sunmaktadır. ESP32-CAM modülü Arduino IDE’de kullanılmak için çeşitli ayarlar yapılmalıdır. Şekil 5.10’da konfigürasyon ayarları gösterilmiştir.



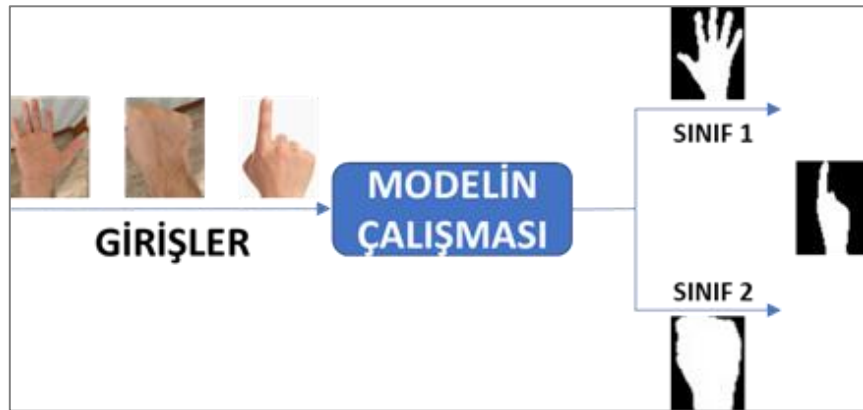
Şekil 5.10. Arduino IDE’de ESP32 konfigürasyon ayarlaması

Kod yükleme işlemi tamamlandığında ESP32-CAM modülü UART (Evrensel Asenkron Alıcı Verici) haberleşme ile sınıflandırma sonucu verileri iletir. UART, iki cihaz arasında seri veri alışverişi için bir protokol veya kurallar kümesi tanımlar. UART, en basit ve en yaygın kullanılan seri iletişim tekniklerinden biridir. Neredeyse tüm mikrodenetleyiciler yerleşik UART donanımlarına sahiptir. Şekil 5.11’de UART haberleşmesi gösterilmiştir.



Şekil 5.11. UART haberleşmesinin şematik gösterimi

Model gerçek zamanlı çalışarak el hareketlerini algılamaktadır. Modelin çalışması Şekil 5.12’de gösterilmiştir. Modelin çalışmasında 2 sınıf bulunmaktadır. Açık el ve kapalı el hareketi olmayan diğer nesneler sınıflandırılmamaktadır. Görsel AKT’de bilinmeyen sınıfı yapılmamıştır. Bunun sebebi 3 sınıf olması durumunda bellek kullanımı artarak ESP32-CAM modeli çalıştıramamıştır.



Şekil 5.12. Görsel AKT modelinin el hareketlerini sınıflandırması

5.2. İşitsel Anahtar Kelime Tespiti

İşitsel anahtar kelime tespiti, ses girdisinden belirli anahtar kelimelerin otomatik olarak tanınması işlemidir. Bu çalışmada Edge Impulse’da eğitilen işitsel AKT modeli Arduino Nano BLE Sense’de çalıştırılmıştır. Arduino BLE Sense, Bluetooth Low Energy (BLE) bağlantısı ve dahili olarak MP34DT05 model bir mikrofon içermektedir. Arduino Nano BLE sense, bir 32-bit Arm cortex-M4 işlemci barındırmaktadır.

5.2.1. İşitsel AKT’de veri toplama

İşitsel AKT’de 3 farklı veri seti kullanılmıştır. İlk veri seti, Google tarafından hazırlanmış olan Yes-No veri setidir [48]. Veri seti yüzlerce farklı kişiden alınan ses kayıtlarından oluşmaktadır. Her bir kelime 1 saniyelik verilerle kaydedilmiştir. Veri seti 342 makalede referans gösterilmiştir [49]. Tablo 5.3’de veri setinin farklı modellerdeki doğruluk yüzdeleri gösterilmiştir. Bu modeller bilgisayar ortamında çalıştırılmış ve AKT uygulamalarında kullanılmıştır. Tablonun son satırında ise bu çalışmadaki Edge Impulse tarafından hesaplanan değeri gösterilmektedir.

Tablo 5.3. Farklı çalışmalarda kullanılan Yes- No veri setinin doğruluk değerleri

Model	Doğruluk (%)
TripletLoss-res15 [50]	98,56
BC-ResNet-8 [51]	98
Wav2KWS [52]	97,9
Res 8 [53]	97,8
KWT-3 [54]	97,49
CNN	95

Doğruluk, makine öğreniminde modelin doğru tahmin ettiği örneklerin oranını ifade etmektedir. Doğruluk, Eşitlik 5.4’de gösterilmiştir.

$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.4)$$

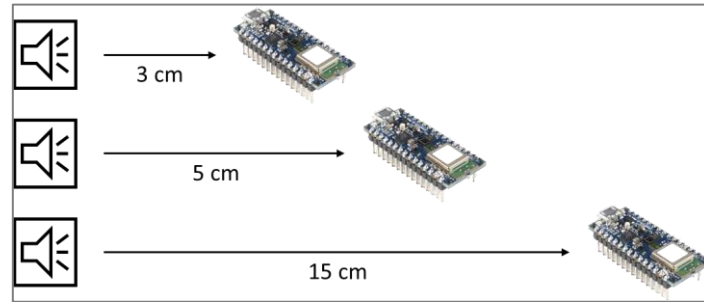
Denklemdaki TN, gerçek negatif örneklerin doğru bir şekilde negatif olarak tahmin edilmesini ifade etmektedir. TP, FP ve FN ifadeleri Eşitlik 5.1 ve Eşitlik 5.2’de açıklanan ifadelerdir.

İkinci veri seti ise Microsoft tarafından hazırlanmış olan MS-SNSD veri setidir [55]. MS-SNSD veri seti, arka plan gürültüsünü bastırmak için derin sinir ağları modellerini eğitmek amacıyla hazırlanmıştır. Bu veri seti çalışmadaki gürültü sınıfında kullanılmıştır.

Üçüncü veri seti ise bu çalışma için Edge Impulse platformunda hazırlanmış olan Open Close veri setidir. Edge impulse platformu kullanıcılara kendi veri setini oluşturma imkânı sunmaktadır. Veriler, modelin çözmeye çalıştığı probleme uygun olmalı ve yeterli miktarda ve çeşitlilikte olmalıdır. Çalışmada Yes-No veri seti hazır olarak alınmıştır ancak yeni bir

veri seti de oluşturulmuştur. Çalışmada MP34DT05 model mikrofon ile 16 kHz örnekleme hızı ile İşitsel veri seti hazırlanmıştır. Her bir kelime 1 saniye olacak şekilde kaydedilmiştir.

Veri seti hazırlanırken konuşmacının konumu önemlidir. Bu çalışmada mikrofona 3 cm, 5 cm ve 15 cm olmak üzere 3 farklı uzaklıktan 3 farklı konuşmacı ile veriler kaydedilmiştir. Veriler 20 dakikalık eğitim ve 5 dakikalık test olmak üzere 25 dakikalıktır. Eğitim verisi %80, test verisi %20 olacak şekilde ayarlanmıştır. Şekil 5.13'de konuşmacıların mikrofona uzaklığı gösterilmiştir.

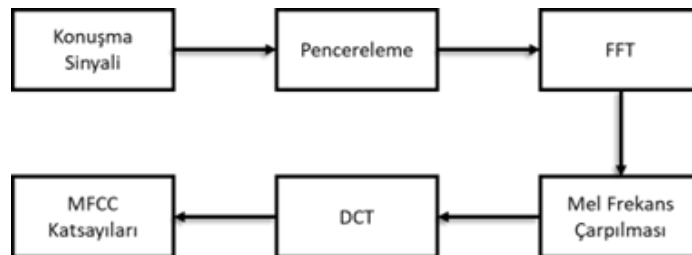


Şekil 5.13. Veri seti oluştururken konuşmacıların mikrofona uzaklığı

5.2.2. İşitsel AKT'de dürtü tasarımı

İşitsel AKT'da dürtü tasarımı adımı, görsel AKT'da olduğu gibidir. İşitsel AKT işleminde, ses sinyalleri dijital bir yapıya dönüştürülüp işlenmektedir.

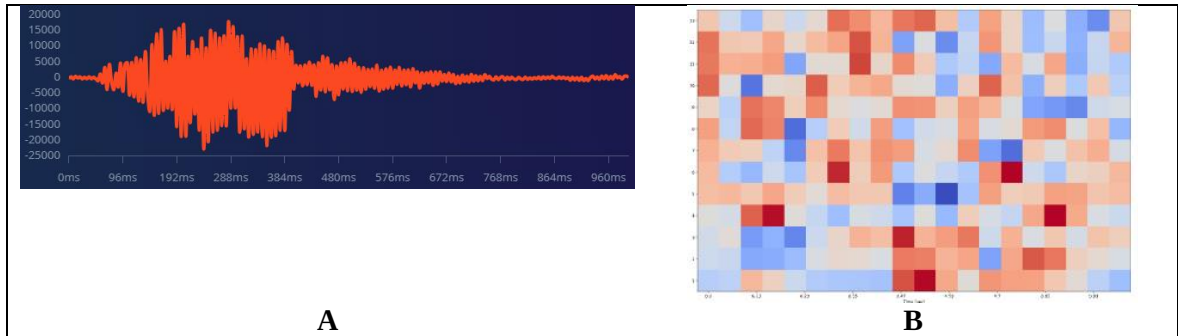
Mel-Frekansı Kepstrum Katsayıları (MFCC), ses sinyallerinin spektral özelliklerini temsil etmek için kullanılan ve insan kulağının frekans algılamasını taklit eden bir gösterim şeklidir. Bu, MFCC'yi konuşma tanıma, ses sentezi, hoparlör doğrulama ve müzik bilgi alma gibi birçok alanda en iyi özelliklerden biri haline getirir. MFCC literatürde de anahtar kelime tespit çalışmalarında kullanılmaktadır [56].



Şekil 5.14. MFCC yapısının blok şeması

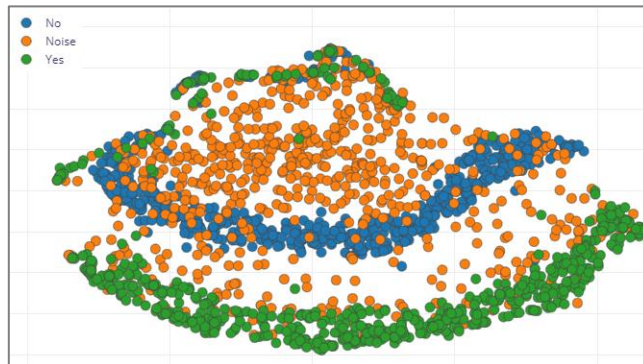
Şekil 5.14’de gösterilen blok şemasında, öncelikle ses sinyalleri analogdan dijitale dönüştürülür. Örnekleme hızı ve bit derinliği belirlenir. Ses sinyali kare pencereler halinde bölünür. Pencerelerin uzunluğu ve örtüşme miktarı belirlenir. Daha sonrasında her pencereye FFT (Hızlı Fourier Dönüşümü) uygulanır. FFT’nin çıktısı, frekans bileşenlerinin genliklerini ve frekanslarını içeren bir spektrumdur. Spektrum, insan işitme sistemine daha yakın bir frekans temsili elde etmek için Mel ölçeği kullanılarak yeniden örnekleme edilir. Mel ölçekli spektrumun DCT (Discrete Cosine Transform)’si hesaplanır. DCT’nin ilk birkaç katsayısı, MFCC’leri oluşturur. En önemli MFCC’ler seçilir. Seçilen MFCC’ler, ses sinyalinin sınıfını tahmin etmek için kullanılır.

Edge impulse, İşitsel verilerini MFCC bloğu şeklinde göstermektedir. MFCC bloğu, İşitsel dalgalarının özelliklerini çıkarmak için kullanılan bir işlem adımını ifade eder. İşlem adımı, İşitsel dalgalarını frekans ve zaman alanında temsil eden bir spektrogramdan başlar. Bu spektrogram daha sonra MFCC’lerin hesaplanması için kullanılır. Şekil 5.15’de işitsel verinin zaman genlik grafiği ve MFCC spektrogramı gösterilmiştir.



Şekil 5.15. İşitsel veri zaman genlik grafiği (A) ve MFCC spektrogramı (B)

Dürtü tasarımı sonucu oluşan özellik gezgini grafiği Şekil 5.16’da gösterilmektedir.



Şekil 5.16. İşitsel sınıflandırma sonucu oluşan özellik gezgini

5.2.3. İşitsel AKT’de eğitim

Görsel verilerin eğitilmesinde açıklandığı gibi eğitim, İşitsel verilerinin yapay sinir ağı modelinin yapılandırılmasını sağlar. Tablo 5.4’de işitsel sınıflandırması eğitim parametreleri gösterilmiştir. Eğitim parametreleri, edge impulse tarafından önerilen veri setine göre en uygun parametrelerdir.

Tablo 5.4. İşitsel sınıflandırmada eğitim parametreleri

Parametre	Miktarı / Durumu
Eğitim Döngüsü	100
Öğrenme Oranı	0.0005
Veri Artırımı	Yok
Dropout	0.25

İşitsel anahtar kelime tespitinde, MFCC ile ses özellikleri çıkarılıp eğitim modeli olarak tek katmanlı CNN yapısı kullanılmıştır. Eğitim sonucunda karışıklık matrisi Şekil 5.17’de gösterilmiştir.

		GERÇEK		
		NO	GÜRÜLTÜ	YES
TAHMIN	NO	99.2%	0%	0.8%
	GÜRÜLTÜ	1.4%	98.2%	0.5%
	YES	0.4%	1.6%	98.0%
	F1SKORU	0.99	0.98	0.98

Şekil 5.17. İşitsel AKT eğitimi sonrası Edge Impulse’da hesaplanmış karışıklık matrisi

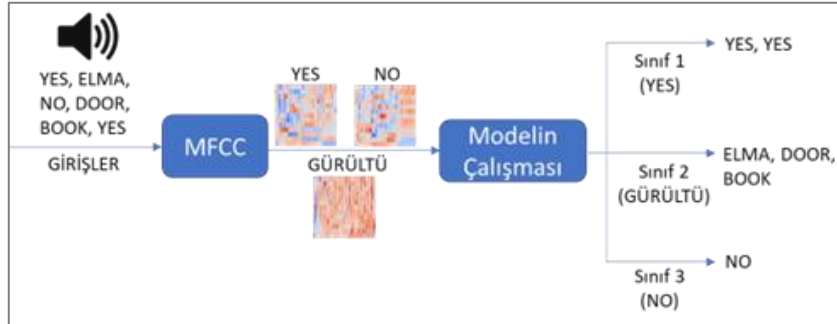
5.2.4. İşitsel AKT’de dağıtım

Dağıtım işleminde Edge Impulse bir kütüphane dosyası olarak model çıktısını vermektedir. Bu kütüphane Arduino IDE programına eklenerek kod cihaza yüklenir. Görsel AKT’de olduğu gibi işitsel AKT için niceleme işlemi Tablo 5.5’de gösterilmiştir.

Tablo 5.5. İşitsel AKT’de optimizasyon sonucu Edge Impulse tarafından hesaplanan değerler

TEKNİK	RAM (kBYTE)	FLASH RAM (kBYTE)	GECİKME (MS)
Niceleme uygulanmış model	13	35,7	340
Niceleme uygulanmamış model	28,5	39,4	981

Model gerçek zamanlı çalışarak sesleri sınıflandırmaktadır. Ses sınıflandırmada 3 sınıf bulunmaktadır. Ses verileri görsel verilere göre daha az yer kapladığı için bu model 3 farklı sınıfı sınıflandırmaktadır. Modelin çalışması Şekil 5.18’de gösterilmiştir.

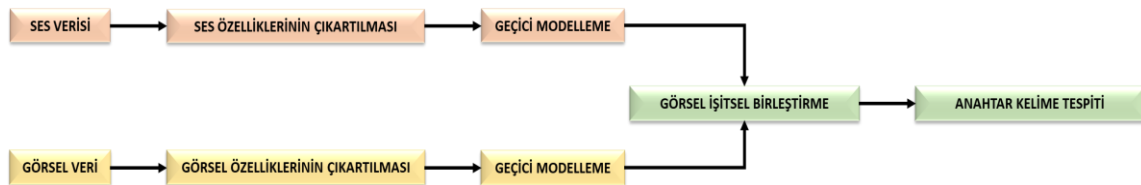


Şekil 5.18. İşitsel AKT modelinin sesleri sınıflandırması

5.3. Görsel İşitsel Anahtar Kelime Tespiti

Görsel-işitsel anahtar kelime tespiti hem görsel hem de işitsel verilerden anahtar kelimeleri otomatik olarak çıkarma ve tanımlama işlemidir. Gürültülü ortamlarda anahtar kelime tespiti yapılması gereken durumlarda kullanılmaktadır. Görsel işitsel anahtar kelime tespiti, makine öğreniminde önemli ve zorlu bir araştırma konusudur. Bu hibrit modeller her iki veri kaynağının sunduğu özellikleri kullanarak daha güçlü ve güvenilir tahminler yapabilir.

Xu ve arkadaşları (2021), uzak alanlarda kelime tespitinin çevresel sağlamlığını iyileştirmek için hem ses hem de görsel bilgiden yararlanarak bir model geliştirmişlerdir. Çalışmada ilk başta çoklu mikrofona sesini ele alıp ses özelliklerini çıkarmak için WPE (Weighted Prediction Error) kullanılmıştır. Görsel verilerin özellik çıkarımı için ROI (Region of Interest) metodu kullanılmıştır. Ardından çok katmanlı CNN ile modeller eğitilmiştir. Modeller doğrusal ağırlıklı birleştirme yapıldıktan sonra %91 doğruluk elde edilmiştir [57]. Görsel işitsel AKT blok şeması Şekil 5.19’da gösterilmiştir.



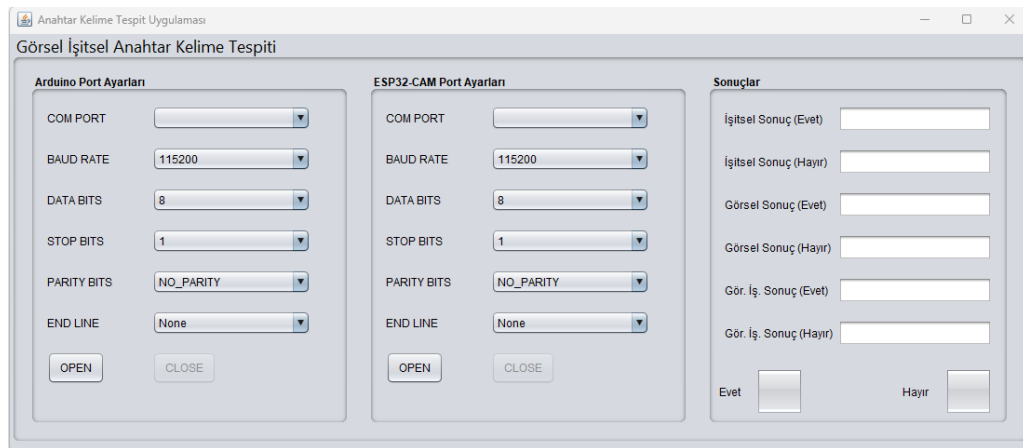
Şekil 5.19. Görsel işitsel AKT blok şeması

Bu tezde, Xu ve arkadaşlarının yapmış olduğu çalışmayı referans alınarak ses için Yes-No ve görüntü için Hand Gesture Recognition Dataset kullanılarak önerilen formül üzerinden modeller birleştirilmiştir. Ayrıca Xu ve arkadaşları bu çalışmayı bilgisayar ortamında yapmışken mevcut çalışma Arduino Nano BLE sense ve ESP32-CAM modüllerinde yapılmıştır.

Donanımlardan ayrı ayrı alınan sonuçlar Eşitlik 5.5 kullanılarak doğrusal ağırlıklı birleştirme yapılmıştır.

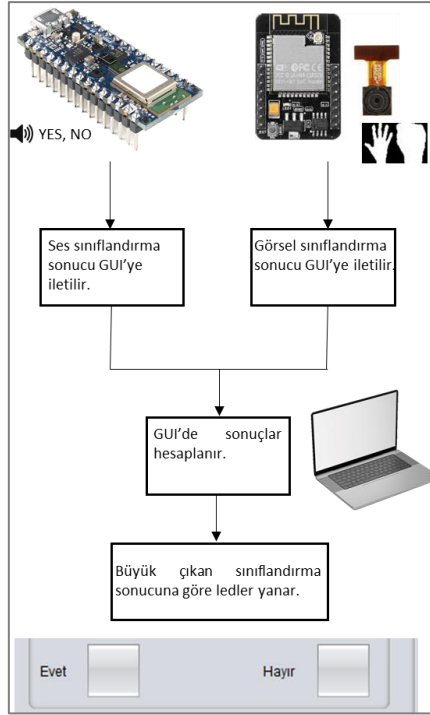
$$y = (w_a \times y_a) + (w_b \times y_b) \quad (5.5)$$

Yukarıdaki denklemdeki w_a ve w_b ifadeleri ağırlık değerleridir. Ağırlık değerleri olarak referans çalışmada w_a için 0.7 ve w_b için 0.3 önerilmiştir. Ayrıca y_a ses modelinin doğruluğunu ve y_b görsel modelin doğruluğunu temsil etmektedir. Model sonuçları, bu çalışma için tasarlanmış JAVA tabanlı bir grafiksel arayüze aktarılmaktadır. Donanımlar ve bilgisayar arasındaki haberleşme UART ile yapılmaktadır. Şekil 5.20’de grafiksel arayüz gösterilmektedir.



Şekil 5.20. JAVA tabanlı tasarlanan grafiksel arayüz

Görsel AKT sonucu 809 milisaniye aralıkla gelmektedir. Ses AKT sonucu ise 340 milisaniye aralıkla gelmektedir. Bu yüzden verilerin sınıflandırılması çıkarım zamanı büyük olan sisteme göre yapılmıştır. Bu işlemde öncelikle ses verisi geldiğinde hafızaya atılıp görsel sonuç beklenmektedir. Görsel sonuç geldiğinde ise sonuçlar Eşitlik 5.5 ile hesaplanmıştır. Büyük çıkan doğruluk sonucuna göre ilgili led yanmaktadır. Şekil 5.21’de görsel işitsel AKT modelinin çalıştırılması gösterilmiştir.



Şekil 5.21. GUI ortamında görsel işitsel AKT modelinin çalıştırılması

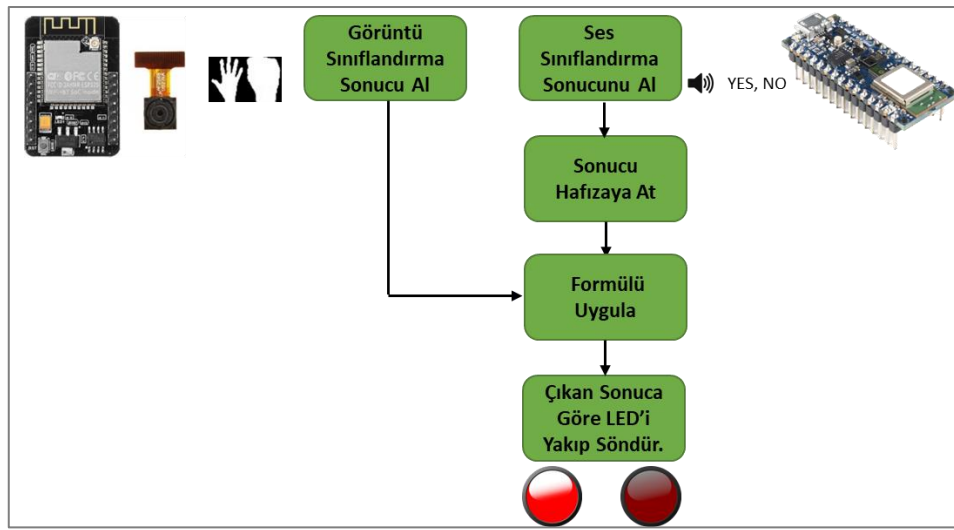
Eşitlik 5.5 değeri ilgili sınıfta 0.7 (%70) üzeri olduğunda ise ilgili sınıfın ledi ile sonuç gösterilmiştir. Şekil 5.22’de GUI ortamında AKT sonuçları gösterilmiştir.

Sonuçlar		Sonuçlar	
İşitsel Sonuç (Evet)	0.86	İşitsel Sonuç (Evet)	0.00
İşitsel Sonuç (Hayır)	0.00	İşitsel Sonuç (Hayır)	0.92
Görsel Sonuç (Evet)	0.99063	Görsel Sonuç (Evet)	0.01641
Görsel Sonuç (Hayır)	0.00938	Görsel Sonuç (Hayır)	0.98281
Gör. İş. Sonuç (Evet)	0.88	Gör. İş. Sonuç (Evet)	0
Gör. İş. Sonuç (Hayır)	0.02	Gör. İş. Sonuç (Hayır)	0.94
Evet ☒	Hayır ☐	Evet ☐	Hayır ☒

Şekil 5.22. GUI ortamında AKT sonuçları

TinyML uygulamaları düşük güç tüketimine ve düşük maliyetli sistemlerde çalışmayı hedeflemektedir. Çalışmanın bilgisayar ortamından bağımsız bir şekilde de çalışması

sağlanmıştır. Burada da öncelikle ses sınıflandırma sonucu alınıp bir hafızaya atılmıştır. Daha sonrasında görsel sınıflandırma sonucuna göre ESP32-CAM modülündeki GPIO14 pini aktif olmaktadır. İlgili sınıflandırma doğruluğu %70 (0,7) üzerinde olduğunda GPIO14 pininden voltaj çıkışı verilmektedir. ESP32-CAM’de açık el hareketi algılandığında GPIO14 pininden 24V çıkış vermektedir. GPIO14 pini Arduino’daki A0 pinine bağlıdır. A0 pini burayı sürekli olarak okumaktadır. Ses sonucu “Yes” olarak algılanıp A0 pininden de 24V okunduğunda Arduino Nano BLE Sense’deki dahili led yanmaktadır. Ses sonucu “No” olarak algılanıp A0 pininden de 0V okunduğunda Arduino Nano BLE Sense’deki dahili led sönmektedir. Şekil 5.23’de donanım ortamındaki modelin çalıştırılması gösterilmiştir.



Şekil 5.23. Donanım ortamında görsel işitsel AKT

6. DENEYLER VE TARTIŞMA

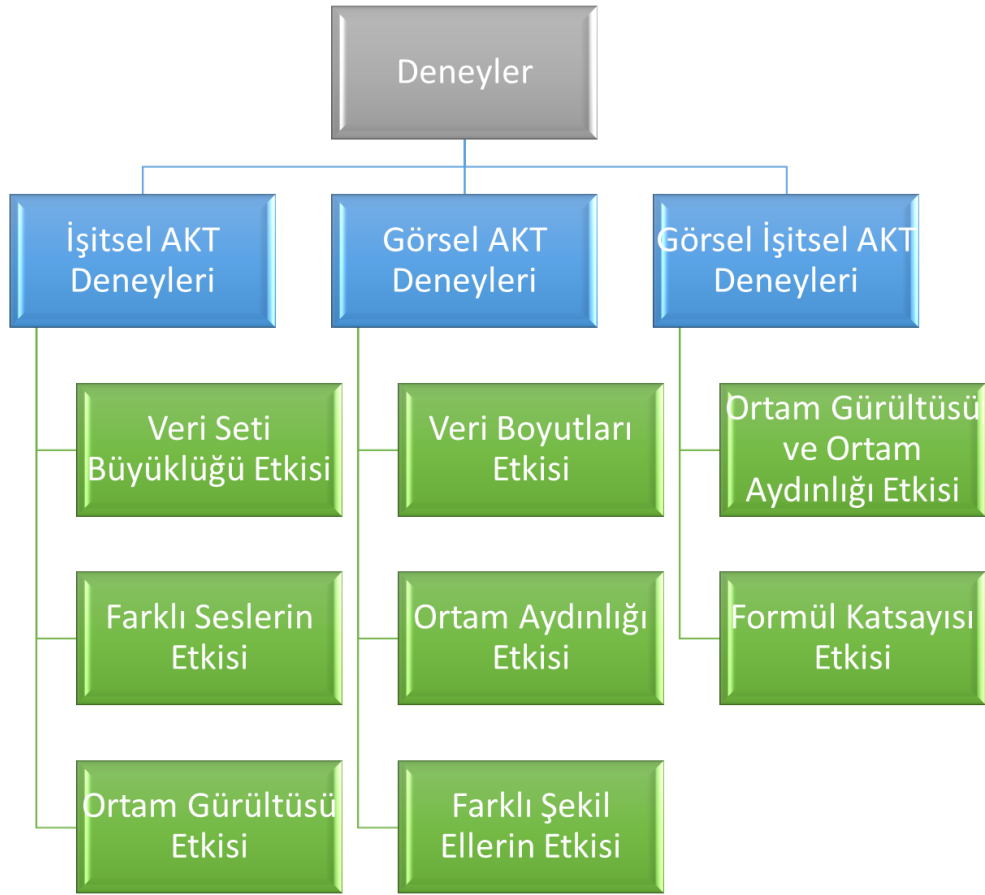
Bu bölümde, edge impulse platformunda geliştirilen ve Arduino IDE ortamında cihazlara yüklenen TinyML uygulamalarının ayrı ayrı görsel ve işitsel deneyleri ve daha sonra yapılan GUI üzerinde görsel işitsel anahtar kelime tespiti deneyleri açıklanmıştır.

Şekil 6.1’de AKT modellerine yapılan deneyler gösterilmektedir. Deneylerdeki amaç görsel AKT ve işitsel AKT modellerinin tek başına değişen fiziksel koşullardaki performansını gözlemlemektir. Daha sonrasında hibrit modelinde performansı test edilip sonuçları karşılaştırarak hibrit modelin başarısını gözlemlemektir.

İşitsel AKT deneyleri 3 aşamada yapılmıştır. Öncelikle veri seti büyüklüğüne göre modelin doğruluk performansı ve aşırı öğrenme (overfitting) durumu gözlemlenmiştir. Daha sonrasında veri setinde olmayan insan sesleriyle model test edilmiştir. Son olarak gürültülü ortamlarda işitsel AKT’nın başarısı gözlemlenmiştir.

Görsel AKT deneylerinde ilk olarak görsel verilerin boyutlarının bellek kullanımı, doğruluk gibi parametre etkisi gözlemlenmiştir. Daha sonrasında ortam aydınlığının modelin performansına etkisi gözlemlenmiştir ve son olarak veri setinde bulunmayan farklı el şekillerinin model performansına etkisi gözlemlenmiştir.

Görsel işitsel AKT deneylerinde ise Eşitlik 5.5’de verilen doğrusal birleştirme yöntemiyle oluşturulan hibrit modelin formül katsayısı ve gürültülü ortamdaki performansı gözlemlenmiştir.



Şekil 6.1. Çalışmada yapılan deneyler

6.1. İşitsel AKT Deneyleri

Görsel İşitsel AKT deneyleri sonucunda gürültülü bir ortamda %85 doğruluk elde edilmiştir.

6.1.1. İşitsel AKT’de veri seti büyüklüğü etkisi

Yapılan çalışmada, değişen veri seti büyüklüğü referans alınarak, doğruluk değeri ve Arduino geliştirme kartı üzerinde yapılan testlerin deneysel doğruluğu gösterilmiştir.

Veri seti olarak Edge Impulse sitesinde Arduino’daki dahili mikrofon ile hazırlanan Open-Close ve hazır olarak alınan Yes-No ile deneyler ayrı ayrı yapılmıştır. Test sütunundaki değer Arduino’dan alınan 20 örnek sonucunda alınan doğruların yüzdesidir. Tablo 6.1’de Open-Close veri setinin büyüklüğünün test ve doğruluğa etkisi

gösterilmektedir. Tablo 6.2’de Yes-No veri setinin büyüklüğünün test ve doğruluğa etkisi gösterilmektedir.

Tablo 6.1. Open-Close veri seti büyüklüğünün doğruluğa etkisi

EĞİTİM VERİSİ (DAKİKA)	TEST VERİSİ (DAKİKA)	DOĞRULUK (%)	TEST (%)
5	1,25	100	35 (20/8)
10	2,5	100	60 (20/12)
20	5	100	75 (20/15)

Tablo 6.2. Yes-No veri seti büyüklüğünün doğruluğa etkisi

EĞİTİM VERİSİ (DAKİKA)	TEST VERİSİ (DAKİKA)	DOĞRULUK (%)	TEST (%)
5	1,25	93,9	55 (20/11)
10	2,5	96,4	80 (20/16)
20	5	98,5	95 (20/19)

Open-Close veri seti 3 farklı insan sesi kullanarak hazırlanmıştır. Yes-No veri seti ise daha profesyonel olarak yüzlerce kişi tarafından Google destekli hazırlanmıştır. Deneylerde Open-Close veri setinin yeterli dengede hazırlanamadığı ve aşırı öğrenme durumu oluşturduğu gözlemlenmiştir. Diğer deneyler daha dengeli ve daha yüksek doğruluk değeri elde ettiği için Yes-No veri seti ile yapılmıştır.

6.1.2. İşitsel AKT’de farklı insan seslerinin etkisi

Bir işitsel AKT modelinin yüksek performans kriterlerinden birisi de farklı seslerde benzer performansı göstermesidir. Tablo 6.3’de 10 farklı insan sesi ile modelin test edilmesini göstermektedir.

Tablo 6.3. Farklı insan seslerinin test doğruluğuna etkisi

CİNSİYET	YAŞ	TEST (%)
ERKEK	28	95 (20/19)
KADIN	51	90 (20/18)
ERKEK	51	85 (20/18)
ERKEK	30	95 (20/19)
ERKEK	40	90 (20/18)
ERKEK	23	90 (20/18)
KADIN	32	95 (20/19)
ERKEK	28	90 (20/18)
KADIN	21	95 (20/19)
KADIN	22	95 (20/19)

Görüldüğü gibi sonuçlar benzer çıkarak modelin farklı seslerde de benzer performans verdiği gözlemlenmiştir.

6.1.3. İşitsel AKT’de ortam gürültüsü etkisi

Ortam gürültüleri genel olarak modelin performansını bozan etkilere sahiptir. Yüksek performans gösteren bir modelin ortam gürültüsünde de başarılı bir performans göstermelidir.

Tablo 6.4. Ortam gürültüsünün test doğruluğuna etkisi

ORTAM GÜRÜLTÜSÜ	TEST (%)
Beyaz Gürültü	90 (20/18)
Siyah Gürültü	90 (20/18)
Kahve Dükkânı Arka Plan Gürültüsü	85 (20/17)
Mutfak Arka Plan Gürültüsü	90 (20/19)
Piano Sesi	90 (20/18)
Boş Laf Gürültü	85 (20/18)
Komşu Sesi Gürültü	85 (20/17)

Tablo 6.4’de görüldüğü gibi arka plan gürültüleri doğruluk değerini düşürmüştür. Arka planda insan sesi olan gürültü değerleri diğer gürültülere göre doğruluğu daha azaltmıştır.

6.2. Görsel AKT Deneyleri

Görsel AKT deneyleri sonucunda yeterli aydınlığa sahip bir odada %85 doğruluk elde edilmiştir.

6.2.1. Görsel AKT’de veri boyutları etkisi

Görsel verilerin boyutları, model doğruluğu, RAM kullanımı ve çıkarım zamanını etkileyebilir. Büyük boyutlu veriler genellikle daha fazla bilgi sağlar, ancak modelin eğitimi ve dağıtımı için daha fazla bellek ve işlem gücü gerektirebilir.

Tablo 6.5. Görsel veri boyutlarının doğruluk, RAM ve çıkarım zamanına etkisi

GENİŞLİK (mm)	BOY (mm)	DOĞRULUK (%)	RAM (kBYTE)	ÇIKARIM ZAMANI (MS)	TEST (%)
32	32	100	583,4	394	98,17
48	48	100	584,5	588	100
64	64	100	586,5	645	100
96	96	100	590,6	2231	100
160	160	100	721,6	4084	100

Tablo 6.5’de görüldüğü gibi 48x48 boyutta en iyi sonucu vermiştir. En iyi sonuç en yüksek doğruluk, en düşük RAM kullanımı ve en düşük çıkarım zamanına göre belirlenmiştir. Edge Impulse görüntü verileri için en düşük 32x32 boyutuna izin vermektedir.

6.2.2. Görsel AKT’de ortam aydınlığı etkisi

Ortam aydınlığı görsel AKT’de performansı etkileyen faktörlerden biridir. Ortamın yeterli düzeyde aydınlık olması test performansını olumlu yönde etkilemektedir. Tablo 6.6’da yeterli ve yetersiz ışıklandırılmış bir odada ki görsel AKT performansı gösterilmiştir. Alınan 20 örnek sonucunda doğru sınıflandırmaya göre yüzde belirlenmiştir.

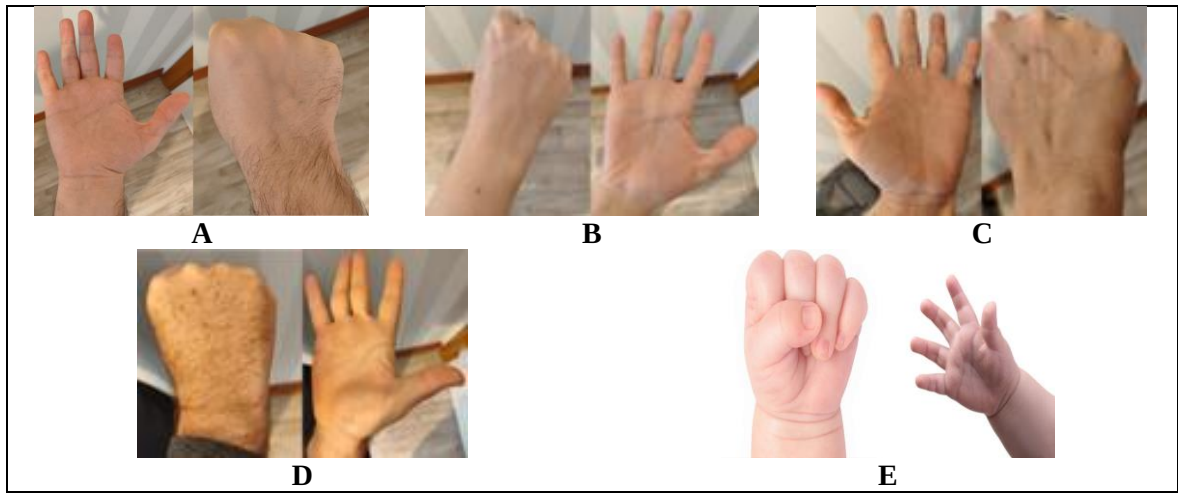
Tablo 6.6. Görsel AKT’de ortam aydınlığının test doğruluğuna etkisi

Ortam Aydınlığı	Test (%)
Yeterli Işıklandırılmış Oda	85 (20/17)
Yetersiz Işıklandırılmış Oda	70 (20/14)

Testler aynı odada yapılmıştır. Yeterli ışıklandırılma, öğle vakti lamba açık bir oda iken, yetersiz ışıklandırılma, akşamüstü lamba açık olmayan bir odadır. Test sütunu, yapılan 20 hareket sonucu doğruluk yüzdesidir.

6.2.3. Görsel AKT’de farklı şekildeki ellerin etkisi

Başarılı bir AKT’de beklenen bir başka durum, farklı boyut ve şekillerdeki ellerde de benzer performansı göstermesidir. Tablo 6.6’da Şekil 6.2’deki (A) eli ile deney yapılmıştır. Tablo 6.7’de ise Şekil 6.2’deki tüm eller ile deneyler yapılmıştır.



Şekil 6.2. Görsel AKT testlerinde kullanılan eller

Tablo 6.7. Görsel AKT’de ortam aydınlığının test doğruluğuna etkisi

El Tipi	Test Performansı (%)
(A)	85 (20/17)
(B)	80 (20/16)
(C)	85 (20/17)
(D)	80 (20/16)
(E)	60 (20/12)

Tablo 6.7’de görüldüğü gibi bebek eli haricinde sonuçlar benzer çıkmıştır. Bebek eli ile yapılan deneylerde test yüzdesi diğer deneylerden düşük çıkmıştır. Bunun sebebi kullanılan veri setinin yetişkin bir el ile hazırlanmasıdır.

6.3. Görsel İşitsel AKT Deneyleri

Bu bölümde Şekil 5.20'deki GUI ile görsel işitsel AKT deneyleri gerçekleştirilmiştir. Deneylerdeki değerlendirme kriteri, görsel işitsel AKT performansının işitsel AKT performansından daha yüksek olmalıdır.

6.3.1. Görsel İşitsel AKT'de ortam aydınlığının etkisi

Tablo 6.8'de farklı ortam şartlarına göre görsel işitsel kelime tespit uygulaması yapılmıştır. Gürültülü ortam, komşu sesi gürültüsü açılmış bir odadır. Yeterli ışıklandırılmış ortam, öğle vakti lamba açık bir odayı temsil ederken, yetersiz ışıklandırılmış ortam akşamüstü lamba kapalı bir odayı temsil etmektedir. Testlerde bazı durumlarda işitsel ve görsel sınıflandırmalar ters yapılmıştır. Örneğin AÇ hareketi yaparken NO sesi söylenmiş veya KAPAT hareketi yaparken YES sesi söylenerek gözlemlenmiştir. Eşitlik 6.5 değerleri olarak w_a için 0.7 ve w_b için 0.3 kullanılmıştır.

Tablo 6.8. Görsel işitsel AKT'de ortam aydınlığının etkisi

Ortam	Görsel İşitsel Algılama (%)	İşitsel Algılama (%)
Yeterli Işık / Sessiz	95	95
Yeterli Işık / Gürültülü	90	85
Yetersiz Işık / Sessiz	95	95
Yetersiz Işık / Gürültülü	80	85

Testler sonucunda yeterli ışık alan gürültülü bir ortamda görsel işitsel algılama işitsel algılama doğruluğundan yüksek çıkmıştır.

6.3.2. Görsel işitsel AKT'de formül katsayılarının etkisi

Referans çalışmada katsayı değerleri w_a için 0.7 ve w_b için 0.3 kullanılmıştır. Yapılan deneylerde w_a katsayısının daha yüksek olması durumunda daha iyi sonuç verildiği gözlemlenmiştir. Tablo 6.9'da w_a katsayısının daha yüksek olduğu diğer durumlar gözlemlenmiştir. Ortam olarak gürültülü ve yeterli ışık alan bir ortam tercih edilmiştir.

Tablo 6.9. Görsel işitsel AKT’de farklı katsayıların etkisi

Katsayılar	Görsel İşitsel Algılama (%)	İşitsel Algılama (%)
$w_a = 0,6 - w_b = 0,4$	85	85
$w_a = 0,7 - w_b = 0,3$	90	85
$w_a = 0,8 - w_b = 0,2$	85	85
$w_a = 0,9 - w_b = 0,1$	85	85

Yapılan deney sonucunda referans çalışmanın önerdiği katsayının en yüksek doğruluk sonucunu verdiği gözlemlenmiştir.

7. SONUÇLAR

Bu çalışma, TinyML ile görsel ve işitsel AKT sonuçlarının entegrasyonunu ele alarak, görsel işitsel anahtar kelime tespiti konusunda önemli bir katkı sağlamıştır.

Öncelikle görsel ve işitsel AKT modelleri Edge Impulse platformunda eğitilip cihazlarda ayrı ayrı test edilmiştir. Görsel AKT yeterli ışık alan bir odada %85 doğruluk elde ederken, işitsel AKT gürültülü bir ortamda %85 doğruluk elde edilmiştir. Daha sonrasında tasarlanan GUI'de ses ve görüntü komutlarını eş zamanlı olarak alarak, doğrusal ağırlıklı birleştirme yapılmıştır. Görsel işitsel anahtar kelime tespiti ile gürültülü bir ortamda %90 doğruluk sonucu elde edilmiştir.

Çalışma, Xu ve arkadaşlarının yapmış olduğu çalışmayı referans alarak farklı veri seti ve eğitim modeli ile gömülü cihazlarda uygulanmıştır. Referans çalışmada hazırlanan veri seti ve kullanılan modeller ile %91 oranında bir başarı sağlamışlardır. Sonuç olarak, modelin küçülmesi ve daha az kapasite kullanmasına rağmen doğruluk sonuçları benzer çıkmıştır. Önerilen yöntemin gerçek zamanlı sistemlerde düşük güç tüketimi ve düşük bellek gereksinimi gibi pratik avantajları, endüstrideki kullanılabilirliğini artırmaktadır.

Çalışmamız, düşük güçlü cihazlarda gerçek zamanlı anahtar kelime tespiti için TinyML'nin potansiyelini göstermektedir. Ancak, gelecekteki çalışmalar için birkaç önemli husus bulunmaktadır. Görsel işitsel yöntemlerin birleştirilmesinin en önemli kullanım alanı robot-insan ve insan makine iletişimi olacaktır. TinyML açısından bu uygulamanın geliştirilmesi küçük robotlar ve taşınabilir cihazlar ve insan iletişimi olabilir. Ayrıca bu çalışmada görsel ve işitsel veriler veri setlerinden ayrı ayrı alınarak fotoğraf ve ses tabanlı çalışmaktadır. İleri çalışmalarda bir videodan otomatik olarak görsel ve işitsel veriler alınarak model çalıştırılabilir.

KAYNAKLAR

- [1] R. K. Varshney, S. Pratap Singh Chauhan and V. Sharma, "Perspectives on the Impact of Artificial Intelligence & Machine Learning on Processes & Structures Engineering," *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022*, pp. 747-752, doi: 10.1109/ICAC3N56670.2022.10074254.
- [2] H. Dong, "The Advance of the Combination Method of Machine Learning and Deep Learning," *2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI), Zakopane, Poland, 2022*, pp. 350-353, doi: 10.1109/ICDACAI57211.2022.00076.
- [3] Feng-Qin Li and Ya-Dong Wu, "Multi-keyword spotting based on speech feature space trace matching," *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826), Shanghai, China, 2004*, pp. 3542-3546 vol.6, doi: 10.1109/ICMLC.2004.1380403.
- [4] Z. Zhang, "Construction of English Teaching System Based on Deep Learning Models and Artificial Intelligence Technology," *2023 2nd International Conference on Artificial Intelligence and Computer Information Technology (AICIT), Yichang, China, 2023*, pp. 1-5, doi: 10.1109/AICIT59054.2023.10277704.
- [5] R. Tang and J. Lin, "Deep Residual Learning for Small-Footprint Keyword Spotting," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018*, pp. 5484-5488, doi: 10.1109/ICASSP.2018.8462688.
- [6] M. Laxmi, B. Sreelatha, V. G and M. Sowjanya, "Enhancing Keyword Spotting in Noisy Environments: A Deep Learning Approach with MFCC and Bidirectional LSTM Networks," *2023 International Conference on System, Computation, Automation and Networking (ICSCAN), PUDUCHERRY, India, 2023*, pp. 1-6, doi: 10.1109/ICSCAN58655.2023.10395591.

- [7] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki and A. S. Hafid, "A Comprehensive Survey on TinyML," in *IEEE Access*, vol. 11, pp. 96892-96922, 2023, doi: 10.1109/ACCESS.2023.3294111.
- [8] Pete Warden, Daniel Situnayake, "TinyML Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers", *O'Reilly Media*, 2019.
- [9] Yelchuri, Harsha & R, Rashmi. (2022). A review of TinyML. 10.48550/arXiv.2211.04448.
- [10] Partha Pratim Ray, "A review on TinyML: State-of-the-art and prospects", *Journal of King Saud University – Computer and Information Sciences* 34 (2022) 1595–1623, 30 November 2021.
- [11] Visal Rajapakse, Ishan Karunanayake, and Nadeem Ahmed. 2023. Intelligence at the Extreme Edge: A Survey on Reformable TinyML. *ACM Comput. Surv. Just Accepted (February 2023)*. doi: 10.1145/3583683.
- [12] Alajlan, N. N., & Ibrahim, D. M. (2022). TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications. *Micromachines*, 13(6), 851. MDPI AG. doi: 10.3390/mi13060851.
- [13] Soro, S. (2021). TinyML for ubiquitous Edge AI. arXiv preprint arXiv:2102.01255.
- [14] Schizas N, Karras A, Karras C, Sioutas S. TinyML for Ultra-Low Power AI and Large Scale IoT Deployments: A Systematic Review. *Future Internet*. 2022; 14(12):363. doi:10.3390/fi14120363.
- [15] Boz, M. & Durgun, Y. (2023). Parkinson hastalarının aktivitelerinin tanınmasında TinyML tabanlı uç bilişim sistemi. *Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*, 12 (1), 317-323. DOI: 10.28948/ngumuh.1144946.
- [16] Mansoureh Lord, TinyML Anomaly Detection, Thesis, scholarworks: 10211.3/219966.
- [17] Marc MONFORT GRAU, "TinyML From Basic to Advanced Applications", *Facultat d'Informatica de Barcelona*, June 21, 2021.

- [18] Moez Altayeb, Marco Zennaro, Ermanno Pietrosemoli, "TinyML Gamma Radiation Classifier", *Nuclear Engineering and Technology*, Volume 55, Issue 2, 2023, Pages 443-451, ISSN 1738-5733, doi: 10.1016/j.net.2022.09.032.
- [19] Giovanni Delnevo, Silvia Mirri, Catia Prandi, Pietro Manzoni, "An evaluation methodology to determine the actual limitations of a TinyML-based solution", *An evaluation methodology to determine the actual limitations of a TinyML-based solution*, *Internet of Things*, Volume 22, 2023, 100729, ISSN 2542-6605, doi:10.1016/j.iot.2023.100729.
- [20] Azevedo, Mariana B. and Medeiros, Thaís de A. de and Medeiros, Morsinaldo and Silva, Ivanovitch and Costa, Daniel G., Detecting Face Masks Through Embedded Machine Learning Algorithms: A Transfer Learning Approach for Affordable Microcontrollers. Available at SSRN: ssrn:4404567 or doi:10.2139/ssrn.4404567.
- [21] Ali Reza Sajun, Salsabeel Shapsough, Imran Zuolkernan, Rached Dhaouadi, Edge-based individualized anomaly detection in large-scale distributed solar farms, *ICT Express*, Volume 8, Issue 2, 2022, Pages 174-178, ISSN 2405-9595, doi:10.1016/j.icte.2021.12.011.
- [22] David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., ... & Rhodes, R. (2021). Tensorflow lite micro: Embedded machine learning for tinyml systems. *Proceedings of Machine Learning and Systems*, 3, 800-811.
- [23] N. L. Giménez, F. Freitag, J. Lee and H. Vandierendonck, "Comparison of Two Microcontroller Boards for On-Device Model Training in a Keyword Spotting Task," *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, 2022, pp. 1-4, doi: 10.1109/MECO55406.2022.9797171.
- [24] Y. Huang, T. Hughes, T. Z. Shabestary and T. Applebaum, "Supervised Noise Reduction for Multichannel Keyword Spotting," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 5474-5478, doi: 10.1109/ICASSP.2018.8462346.

- [25] Z. Tang, L. Chen, B. Wu, D. Yu and D. Manocha, "Improving Reverberant Speech Training Using Diffuse Acoustic Simulation," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 6969-6973, doi: 10.1109/ICASSP40776.2020.9052932.
- [26] J. M. Phillips and J. M. Conrad, "Robotic System Control using Embedded Machine Learning and Speech Recognition," *2022 IEEE 19th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, Marietta, GA, USA, 2022, pp. 214-218, doi: 10.1109/HONET56683.2022.10019106.
- [27] Feng-Qin Li and Ya-Dong Wu, "Multi-keyword spotting based on speech feature space trace matching," *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, Shanghai, China, 2004, pp. 3542-3546 vol.6, doi: 10.1109/ICMLC.2004.1380403.
- [28] T. R. N and R. Gupta, "A Survey on Machine Learning Approaches and Its Techniques;," *2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, Bhopal, India, 2020, pp. 1-6, doi: 10.1109/SCEECS48394.2020.190.
- [29] Pragati Baheti (2021). Supervised and Unsupervised Learning [Differences & Examples]. Available: v7labs.com.
- [30] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in *IEEE Access*, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.
- [31] Ömer Faruk Coşkun (2021). Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme Nedir? Aralarındaki Farklar, Available: ofcskn.com.
- [32] C. Wang et al., "An Interpretable and Accurate Deep-Learning Diagnosis Framework Modeled With Fully and Semi-Supervised Reciprocal Learning," in *IEEE Transactions on Medical Imaging*, vol. 43, no. 1, pp. 392-404, Jan. 2024, doi: 10.1109/TMI.2023.3306781.

- [33] M. Somvanshi, P. Chavan, S. Tambade and S. V. Shinde, "A review of machine learning techniques using decision tree and support vector machine," *2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, India, 2016*, pp. 1-7, doi: 10.1109/ICCUBEA.2016.7860040.
- [34] W. Gong and W. Wang, "Application research of support vector machine in E-Learning for personality," *2011 IEEE International Conference on Cloud Computing and Intelligence Systems, Beijing, China, 2011*, pp. 638-642, doi: 10.1109/CCIS.2011.6045147.
- [35] N. L. Giménez, J. Lee, F. Freitag and H. Vandierendonck, "The Effects of Weight Quantization on Online Federated Learning for the IoT: A Case Study," in *IEEE Access*, vol. 12, pp. 5490-5502, 2024, doi: 10.1109/ACCESS.2024.3349557.
- [36] Smith, J., Johnson, A., & Lee, B. (2023). Niche-based Machine Learning: A Review. *Journal of Machine Learning*, doi: 10.18096/TMP.2023.01.03.
- [37] K. Sun, Y. Yu and Z. Huang, "A generalized pruning algorithm for extreme learning machine," *2015 IEEE International Conference on Information and Automation, Lijiang, China, 2015*, pp. 1431-1436, doi: 10.1109/ICInfA.2015.7279511.
- [38] M. M. H. Shuvo, S. K. Islam, J. Cheng and B. I. Morshed, "Efficient Acceleration of Deep Learning Inference on Resource-Constrained Edge Devices: A Review," in *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42-91, Jan. 2023, doi: 10.1109/JPROC.2022.3226481.
- [39] Matthew Stewart (2020). Tiny Machine Learning: The Next AI Revolution, Available: towardsdatascience.com.
- [40] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016*, pp. 770-778, doi: 10.1109/CVPR.2016.90.

- [41] Erdal, H., & Korucuk, S. (2023). Bütünleşik Dalgacık Dönüşümü-Evrişimsel Sinir Ağları Tabanlı Derin Öğrenme Yaklaşımı ve Borsa Tahmini Üzerine Bir Uygulama. *Karadeniz Sosyal Bilimler Dergisi*, 15(29), 387-404. doi:10.38155/ksbd.1258709.
- [42] Muazzez Buket Darıcı (2020). Evrişimsel Sinir Ağı'nda Evrişim İşlemi (The Convolution Process in Convolutional Neural Network), Available: medium.com.
- [43] L. Ou and K. Zhu, "Identification Algorithm of Diseased Leaves based on MobileNet Model", *2022 4th International Conference on Communications, Information System and Computer Engineering (CISCE), Shenzhen, China, 2022*, pp. 318-321, doi: 10.1109/CISCE55963.2022.9851086.
- [44] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
- [45] Srudeep PA (2020). An Overview on MobileNet: An Efficient Mobile Vision CNN, Available: medium.com.
- [46] Mayank2 (2022). Depth wise Separable Convolutional Neural Networks, Available: geeksforgeeks.org
- [47] Rishabh Arya, Hand Gesture Recognition Dataset, Accessed 14 January 2024, Available: kaggle.com.
- [48] Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition. arXiv preprint doi:10.48550/arXiv.1804.03209.
- [49] Papers with code, Speech Commands, Accessed 2 February 2024, Available: paperswithcode.
- [50] VYGON, Roman; MIKHAYLOVSKIY, Nikolay, Learning efficient representations for keyword spotting with triplet loss, *23rd International Conference, SPECOM 2021*, doi:10.1007/978-3-030-87802-3_69.
- [51] KIM, Byeonggeun, et al. Broadcasted residual learning for efficient keyword spotting. arXiv preprint 2021, doi:10.48550/arXiv.2106.04140.

- [52] D. Seo, H. -S. Oh and Y. Jung, "Wav2KWS: Transfer Learning From Speech Representations for Keyword Spotting," in *IEEE Access*, vol. 9, pp. 80682-80691, 2021, doi:10.1109/ACCESS.2021.3078715.
- [53] Tang, R., Lee, J., Razi, A., Cambre, J., Bicking, I., Kaye, J., & Lin, J. (2020). Howl: A deployed, open-source wake word detection system. arXiv preprint arXiv:2008.09606, doi:10.48550/arXiv.2008.09606.
- [54] BERG, Axel; O'CONNOR, Mark; CRUZ, Miguel Tairum. Keyword transformer: A self-attention model for keyword spotting. arXiv preprint arXiv:2104.00769, 2021, doi:10.21437/Interspeech.2021-1286.
- [55] Chandan K. A. Reddy, Ebrahim Beyrami, Jamie Pool, Ross Cutler, Sriram Srinivasan, Johannes Gehrke. *A scalable noisy speech dataset and online subjective test framework*, 2019, doi:10.48550/1909.08050.
- [56] A. Mahmood and U. Köse, "Speech recognition based on convolutional neural networks and MFCC algorithm", *Adv. Artif. Intell. Res.*, vol. 1, no. 1, pp. 6–12, 2021.
- [57] Xu, Y., Sun, J., Han, Y., Zhao, S., Mei, C., Guo, T., ... & Li, X. (2022, May). Audio-visual wake word spotting system for misp challenge 2021. In *ICASSP 2022-2022 IEEE International Conference on Acoustics*, doi:10.48550/arXiv.2204.08686.