



**KTO KARATAY ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ  
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI  
ELEKTRİK VE BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI**

**ARTIRILMIŞ GERÇEKLİK UYGULAMALARINDA NESNELERİN SESLE  
KONTROL EDİLMESİ**

**Tuba EMEKLİ ÇOBAN**

**Yüksek Lisans Tezi**

**KONYA  
Ocak 2023**

ARTIRILMIŐ GERÇEKLİK UYGULAMALARINDA NESNELERİN SESLE  
KONTROL EDİLMESİ

Tuba EMEKLİ ÇOBAN

KTO Karatay Üniversitesi  
Lisansüstü Eğitim Enstitüsü  
Elektrik Elektronik Mühendisliđi Anabilim Dalı  
Elektrik ve Bilgisayar Mühendisliđi Tezli Yüksek Lisans Programı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Ali ÖZTÜRK

Konya  
Ocak 2023

## BİLDİRİM

Enstitü tarafından onaylanan Yüksek Lisans/Doktora tezimin tamamını veya herhangi bir kısmını basılı veya dijital biçimde arşivleme ve aşağıda belirtilen koşullar dahilinde erişime açma iznini KTO Karatay Üniversitesine verdiğimi bildiririm. Bu izinle, Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak ve gelecekteki çalışmalar (makale, kitap, lisans, patent vb.) için tezimin tamamının veya bir bölümünün kullanım hakları yalnızca bana ait olacaktır.

Tezimin bütünüyle kendi çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Telif hakkı bulunan ve sahiplerinden yazılı izinle kullanılması zorunlu olan kaynakları, yazılı izin alarak kullandığımı ve istenildiğinde izinlerin suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayımlanan “Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge” kapsamında, tezim, aşağıda belirtilen koşullar haricince, YÖK Ulusal Tez Merkezi ve KTO Karatay Üniversitesi Açık Erişim Sisteminde erişime açılır.

Enstitü / Fakülte Yönetim Kurulu kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren 2 yıl ertelenmiştir.<sup>1</sup>

Enstitü / Fakülte Yönetim Kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihimden itibaren ... ay ertelenmiştir.<sup>2</sup>

Tezimle ilgili gizlilik kararı verilmiştir.<sup>34</sup>

18 Ocak 2023

**Tuba EMEKLİ ÇOBAN**

<sup>1</sup> MADDE 6(1) Lisansüstü teze ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir.

<sup>2</sup> MADDE 6(2) Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulunun gerekçeli kararı ile altı ay aşmamak üzere tezin erişime açılması engellenebilir.

<sup>3</sup> MADDE 7(1) Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.

<sup>4</sup> MADDE 7(2) Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

## ETİK BEYAN

KTO Karatay Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Hazırlama ve Yazım Kurallarına uygun olarak Doç. Dr. Ali ÖZTÜRK danışmanlığında tarafımdan üretilen bu tez çalışmasında; sunduğum tüm veri, enformasyon, bilgi ve belgeleri bilimsel etik kuralları çerçevesinde elde ettiğimi, tüm değerlendirme, analiz, bulgu ve sonuçları bilimsel usullere uygun olarak sunduğumu, tez çalışmasında yararlandığım kaynakların tümüne bilimsel normlara uygun biçimde atıfta bulunarak kaynak gösterdiğimi, tezimin/projemin kaynak gösterilen durumlar dışında özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

18 Ocak 2023

---

**Tuba EMEKLİ ÇOBAN**

## TEŐEKKÜR

Tez alıŐmamn tım aŐamalarında, kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren deęerli danıŐman Hocam Sayın Do. Dr. Ali Öztürk'e ve baŐta annem, babam, kardeŐlerim ve eŐim olmak üzere hep yanımda olduklarını hissettiren sevgili aileme iten teŐekkürlerimi sunarım.

18 Ocak 2023

Tuba EMEKLİ OBAN

## ÖZET

Tuba EMEKLİ ÇOBAN

Artırılmış Gerçeklik Uygulamalarında Nesnelerin Sesle Kontrol Edilmesi

Yüksek Lisans

Konya, 2023

Artırılmış gerçeklik (AG), bilgisayar ortamında oluşturulan 3 Boyutlu (3B) sanal nesnelere gerçek dünya görüntüleri ile birleştiren bir teknolojidir. Bu şekilde oluşturulan sahnede sanal nesnelere gerçek zamanlı olarak etkileşim kurmak mümkündür. AG teknolojisi uygulamaları eğitim, savunma, tıp, e-ticaret, imalat, turizm ve daha birçok alanda kullanılmaktadır. AG ortamında ses ile etkileşim, uygulamalarda kullanım kolaylığı sağlar. Bu çalışmada Unity yazılımında Vuforia paketi kullanılarak oluşturulan AG ortamındaki sanal nesnelerin sesle kontrolü sağlanmıştır. Bu amaçla farklı cinsiyetten iki kişiden alınan beş farklı komutun ses kayıtları ile bir veri seti oluşturulmuştur. Ses verilerinden elde edilen Mel-frekans Kepstral Katsayıları (MFKK), Doğrusal Tahmini Kodlama (DTK) ve Mel-spektrogram özellikleri Karar Ağacı (KA), Rastgele Orman (RO) ve Destek Vektör Makineleri (DVM) algoritmalarına uygulanmış ve sonuçlar karşılaştırılmıştır. En iyi doğruluk oranları erkek sesi veri setinde RO için %87, kadın sesi veri setinde DVM için %89 olarak bulunmuştur. İki veri setinin birleşiminden oluşan veri seti için doğruluk oranı hem RO hem de DVM için %85 olarak bulunmuştur. Öte yandan, MFKK özellikleri ile Dinamik Zaman Bükme (DZB) yöntemini kullanan k-En Yakın Komşu (kEYK) algoritmasının doğruluk oranı erkek sesi veri setleri için %89 ve kadın sesi veri setleri için %88 olarak bulunmuştur. İki veri setinin birleşiminden oluşan veri seti için kEYK doğruluk oranı %83 olarak bulunmuştur. Tüm makine öğrenimi algoritmaları için sonuçlar çapraz doğrulama ile elde edilmiştir.

### **Anahtar Kelimeler**

Artırılmış Gerçeklik, Ses Tanıma, Makine Öğrenimi

## **ABSTRACT**

Tuba EMEKLİ ÇOBAN

Voice Control of Objects in Augmented Reality Applications

Master's

Konya, 2023

Augmented reality (AR) is a technology that combines 3 Dimensional (3D) virtual objects created in a computer environment with real world images. In the scene created in this way, it is possible to interact with virtual objects in real time. AR technology applications are used in education, defense, medicine, e-commerce, manufacturing, tourism and many more. Interaction with voice in AR environment provides ease of use in applications. In this study, voice control of virtual objects in the AR environment created by using the Vuforia package in Unity software is provided. For this purpose, a data set was created with voice recordings of five different commands taken from two people of different genders. Mel-frequency Cepstral Coefficients (MFCC), Linear Predictive Coding (LPC) and Mel-spectrogram features obtained from audio data were applied to Decision Tree (DT), Random Forest (RF) and Support Vector Machines (SVM) algorithms and the results were compared. The best accuracy rates were found to be 87% for RF in male voice dataset and 89% for SVM in female voice dataset. The accuracy rate for the data set consisting of the combination of the two data sets was found to be 85% for both RF and SVM. On the other hand, the accuracy of the k-Nearest Neighbor (kNN) algorithm, which uses the Dynamic Time Warping (DTW) method with MFCC features, was found to be 89% for male voice datasets and 88% for female voice datasets. For the data set consisting of the combination of the two data sets, the accuracy rate of kNN was found to be 83%. Results for all machine learning algorithms were obtained by cross validation.

### **Keywords**

Augmented Reality, Voice Recognition, Machine Learning

## İÇİNDEKİLER

BİLDİRİM .....	i
ETİK BEYAN.....	ii
TEŞEKKÜR.....	iii
ÖZET.....	v
ABSTRACT.....	vi
İÇİNDEKİLER .....	vii
TABLolar DİZİNİ .....	ix
ŞEKİLLER DİZİNİ.....	x
SİMGELER DİZİNİ.....	xi
KISALTMALAR DİZİNİ.....	xii
1. GİRİŞ .....	1
2. ARTIRILMIŞ GERÇEKLIK .....	5
2.1. AG İçin Kullanılan Donanım ve Yazılımlar .....	5
2.2. AG Türleri .....	6
2.2.1. Konum Tabanlı Artırılmış Gerçeklik Uygulamaları (KTAG).....	6
2.2.2. Görüntü Tabanlı Artırılmış Gerçeklik Uygulamaları (GTAG).....	6
3. LİTERATÜRDEKİ AG UYGULAMALARI .....	9
3.1. Eğitim Alanındaki Çalışmalar .....	9
3.2. Sanat Alanındaki Çalışmalar .....	13
3.3. Trafik Alanındaki Çalışmalar .....	13
3.4. Doğal Afetlerle İlgili Çalışmalar .....	14
3.5. AG Uygulamalarında Sesin Kullanıldığı Çalışmalar .....	15
4. SES TANIMA .....	17
4.1. Ses Tanıma Adımları.....	18
5. MATERYAL VE YÖNTEMLER .....	20
5.1. Yazılım Paketleri .....	20
5.2. Donanımlar .....	20
5.3. Kullanılan Veri Kümesi.....	20
5.4. Ses Kayıtlarından Özellik Çıkarma .....	21
5.4.1. Mel-Frekans Kepstral Katsayıları (MFKK).....	21
5.4.2. Doğrusal Tahmini Kodlama (DTK) .....	22



5.4.3. Mel-spectrogram .....	22
5.5. Makine Öğrenmesi Algoritmaları.....	23
5.5.1. Karar Ağacı (KA) .....	23
5.5.2. Rastgele Orman (RO) .....	23
5.5.3. Destek Vektör Makineleri (DVM).....	24
5.5.4. k-En Yakın Komşu (kEYK) .....	25
5.5.4.1. Dinamik Zaman Bükme (DZB) .....	25
6. DENEYSEL SONUÇLAR.....	26
7. SONUÇ.....	33
KAYNAKLAR .....	34
ÖZGEÇMİŞ .....	38
EK 1. KULLANILAN KODLAR.....	39

## TABLULAR DİZİNİ

Tablo 1. AG Uygulama Geliştirme Yazılım Araçları .....	6
Tablo 2. kEYK algoritmasının erkek ses veri seti için karmaşıklık matrisi.....	28
Tablo 3. kEYK algoritmasının kadın ses veri seti için karmaşıklık matrisi.....	29
Tablo 4. kEYK algoritmasının birleştirilmiş veri seti için karmaşıklık matrisi.....	29
Tablo 5. Birleşik veri kümesi için DVM algoritmasının karmaşıklık matrisi.....	30
Tablo 6. Birleşik veri kümesi için RO algoritmasının karmaşıklık matrisi .....	31
Tablo 7. Birleşik veri kümesi için KA algoritmasının karmaşıklık matrisi .....	31

## ŞEKİLLER DİZİNİ

Şekil 1. Gerçeklik sanallık düzlemi .....	5
Şekil 2. İbili ve Şahin (2013) tarafından geliştirilen AG uygulamasına ait bir ekran görüntüsü.....	9
Şekil 3. Perez-Lopez ve Contero (2013) tarafından geliştirilen AG uygulamasına ait bir ekran görüntüsü.....	10
Şekil 4. Di Serio vd. (2012) tarafından geliştirilen AG uygulamasına ait ekran görüntüleri.....	10
Şekil 5. Juan vd. (2010) geliştirdiği AG uygulamasına ait bir ekran görüntüsü.....	11
Şekil 6. Enyedy vd. (2016) geliştirdiği AG uygulamasını açıklayan görüntüler.....	12
Şekil 7. Yang ve Liao (2014)'ya ait AG uygulamasını açıklayan görüntüler.....	12
Şekil 8. Bir müzede AG uygulaması ile yaşanan deneyim .....	13
Şekil 9. Trafikle ilgili AG uygulaması ekran görüntüsü.....	14
Şekil 10. "Sola Dön" komutu ses kaydı a) Kadın b) Erkek .....	21
Şekil 11. Bu tez çalışması için geliştirilen AG uygulaması.....	26
Şekil 12. a) "Sola Dön" ses sinyalinin orijinal grafiği, b) Sessizliğin kaldırılmasından sonraki grafiği c) Gürültü filtrelemeden sonraki grafiği.....	27
Şekil 13. a) MFKK değerleri b) Melspec değerleri c) DTK değerleri.....	28
Şekil 14. İlk veri seti üzerinde algoritmaların ortalama doğruluk oranları .....	30
Şekil 15. Genişletilmiş veri seti üzerinde algoritmaların ortalama doğruluk oranları ....	32

## SİMGELER DİZİNİ

Simge	Açıklama
$\Sigma$	Sigma
$p_i$	i özelliğinin olasılığı
$P(y_i)$	$y_i$ sınıfının olasılığı
$\in$	Eleman
$V$	Değerler kümesi
$\pi$	Pi sayısı
$\alpha_i$	Lagrange katsayısı
$\delta$	Uzaklık fonksiyonu

## KISALTMALAR DİZİNİ

<b>Kısaltma</b>	<b>Açıklama</b>
SG	Sanal Gerçeklik
AG	Artırılmış Gerçeklik
3B	3 Boyutlu
3D	3 Dimensional
MFKK	Mel-Frekans Kepstral Katsayı
MFCC	Mel-Frequency Cepstral Coefficient
DTK	Doğrusal Tahmini Kodlama
LPC	Linear Predictive Coding
KEYK	k-En Yakın Komşu
kNN	k Nearest Neighbours
DZB	Dinamik Zaman Bükme
DTW	Dynamic Time Warping
RO	Rastgele Orman
RF	Random Forest
DVM	Destek Vektör Makineleri
SVM	Support Vector Machines
KA	Karar Ağacı
DT	Decision Tree
İT	İşaretçi Tabanlı
İTO	İşaretçi Tabanlı Olmayan
İTAG	İşaretçi Tabanlı Artırılmış Gerçeklik
İTOAG	İşaretçi Tabanlı Olmayan Artırılmış Gerçeklik
KTAG	Konum Tabanlı Artırılmış Gerçeklik
GTAG	Görüntü Tabanlı Artırılmış Gerçeklik
PTAG	Projeksiyon Tabanlı Artırılmış Gerçeklik
STAG	Süperpoze Tabanlı Artırılmış Gerçeklik
GPS	Global Positioning System
WiFi	Wireless Fidelity
RFID	Radio Frequency Identification
FFT	Fast Fourier Transform
GB	Gigabyte

RAM                      Random Access Memory  
CPU                      Central Processing Unit

## 1. GİRİŞ

Gerçeklik teknolojisi sanal gerçeklik (SG) ve AG olarak ikiye ayrılabilir. SG ortamı gerçekçi sanal görüntülerin olduğu, seslerin duyulabildiği veya dokunma gibi bazı duyuların hissedilebildiği ve kullanıcının gerçek dünya ile ilişkisinin algısal olarak yok olduğu bir ortamdır. Google Cardboard (2014), Samsung Gear VR, Google Daydream gibi donanımlar ile SG ortamlarını algılamak için cep telefonu gereklidir. Cep telefonuna ihtiyaç duymayan SG donanımları ise ekran, bazı sensörler ve lensler içerirler ve kullanıcının baş hareketini algırlar. İkinci gruptaki donanımlara örnek olarak Oculus Rift (2013), HTC Vive ve Sony Playstation VR verilebilir. Bu donanımlardaki lensler kullanıcının stereoskopik görüntüleri algılamalarına izin verir. Stereoskopik görüntü tekniği iki boyutlu bir görüntüye 3B efektler vererek derinlik algısı kazandırır.

AG ortamında ise; gerçek dünya nesnelere ile sanal nesnelere ve verilerin kullanıcı tarafından aynı ortamda birlikte algılanması sağlanır. Gerçek dünya nesnelere üzerine sanal nesnelere eklenerek bütünleşik bir görüntü elde edilir. AG uygulamalarının gerçekleştirilebilmesi için kameraya ihtiyaç vardır. SG uygulamalarında ise kamera kullanılmaz. AG için kullanılan kafaya monteli donanımlar arasında Hololens, Meta2 ve HTC View Cosmos Elite sayılabilir.

Kafaya monteli donanımlar yanında cep telefonu, tablet gibi mobil cihazlar da AG uygulamalarının gerçekleştirimi için kullanılmaktadır. Mobil cihazlar AG uygulamaları için gerekli olan kamera yanında ivmeölçer, Global Positioning System (GPS), Wireless Fidelity (WiFi), gyroscope, pusula gibi donanım ve sensörler de içermektedirler. Ayrıca yüksek ana bellek kapasitesi, yüksek işlemci hızı ve grafik işlemci kullanımı gibi özellikleri ile de AG uygulamalarının performans gereksinimlerini karşılayabilmektedirler. Microsoft firmasının ürünü olan kafaya monteli Hololens cihazı da yukarıda belirtilen sensörlere sahiptir. Fakat, Hololens'in fiyatı yaklaşık 3500\$ ve Meta 2'nin fiyatı ise yaklaşık 1500\$ civarındadır. Bu yüzden AG uygulama geliştirme için pahalı seçeneklerdir. Öte yandan, çok daha ucuz fiyatlara mobil telefon ve tablet satın almak mümkün olduğundan mobil cihazlar AG uygulama geliştirme için oldukça ucuz bir seçenek sunmaktadırlar.

AG ve SG uygulamalarının pazar payının önümüzdeki yıllarda giderek artacağı öngörülmektedir. Facebook şirketinin adını Meta olarak değiştirip Metaverse adını verdiği sanal ortamda uygulama geliştirme ve alt yapı hazırlama faaliyetlerine yüksek miktarda parasal yatırım yaptığı bilinmektedir. Uygulama alanlarının yaygın olduğu ve giderek de artacağı tahmin edilen AG konusunda yapılacak çalışmaların önemi büyüktür.

Literatür incelendiğinde AG alanında pek çok çalışmanın yapıldığı görülmektedir. AG uygulamaları konum tabanlı ve görüntü tabanlı olmak üzere iki farklı türe ayrılmaktadır (Cheng ve Tsai, 2013). Konum tabanlı artırılmış gerçeklik (KTAG) uygulamalarında ekranda gerçek nesnelere yanında sanal nesnelere sahne ortasına çıkması GPS, WiFi, Radio Frequency Identification (RFID) veya sensör verilerine bağlıdır (Sırakaya ve Seferoğlu, 2016). Görüntü tabanlı artırılmış gerçeklik (GTAG) ortamında tanımlanan resim, grafik, kare kod, fotoğraf, hareket veya ses gibi nesnelere işaretçi olarak kullanılmakta, işaretçinin cihaz tarafından algılanması ile sanal nesnelere eklenmektedir. AG alanında geliştirilen uygulamaların ve yapılan çalışmaların büyük çoğunluğu görüntü tabanlıdır (Abdüsselam ve Karal, 2012). Akademik alanda yapılan çalışmalarda ise daha çok görüntü tabanlı yöntemlerden biri olan işaretçi tabanlı artırılmış gerçeklik (İTAG) yöntemi tercih edilmiştir. Çoğunlukla ARToolkit ve Vuforia SDK yazılım araçları kullanılmıştır. İşaretçi tabanlı artırılmış gerçeklik (İTAG) uygulamalarında, görüntülenecek sanal nesne ile ilişkilendirilmiş bir işaretçinin sisteme önceden yüklenmesi gerekmektedir (Wojciechowski ve Wojciech, 2013; İbili ve Şahin, 2013). İTAG uygulamasının avantajları, geliştirmenin nispeten daha kolay olması, ek donanımlara (GPS, RFID, vd) ihtiyaç duymaması ve geniş bir uygulama alanına sahip olmasıdır (İçten ve Bal, 2017). İTAG uygulamalarında, sanal nesne ile ilişkilendirilecek işaretçi örneğin kitap kapağı görüntüsü olabilir. İşaretçi tabanlı (İT) uygulamaların geliştirilmesi diğer türlere göre daha kolaydır. İşaretsiz AG uygulamalarında sanal nesnelere görüntülenecek yer sabit değildir. Sanal nesnelere etkileşim, el hareketleri veya ses gibi girdilerle sağlanır. Unity ile birlikte Vuforia SDK yazılımları kullanılarak Android ve IOS tabanlı mobil cihazlarda çalışabilecek AG uygulamaları geliştirilebilmektedir.

AG uygulamalarının ilk kullanım alanlarından biri savaş pilotlarının kasklarında anlık uçuş verilerinin görüntülenmesi şeklinde olmuştur (Livingston vd., 2011).



Daha sonra yapılan AG çalışmaları incelendiğinde eğitim (Tülü ve Yılmaz, 2012; Antonioli vd., 2014; Motejlek ve Alpay, 2021), doğal afet ve felaketlerde hayatta kalma (Tsai vd., 2013), müzik ve sanat (Trujano vd., 2018; Serafin vd., 2017; Petrović, 2020), fen bilimleri (Enyedy vd., 2012), mühendislik (Zaher vd., 2018; Su vd., 2020), engelli insanlar (Yoon vd., 2019), trafik (Zhang, 2016) gibi çok çeşitli alanlarda uygulamalar olduğu görülmektedir.

Endüstriyel alanda yapılan uygulamalara bakıldığında AG ile ilgili çalışmalar yapan şirketler olduğu görülmektedir. Uluslararası alanda GPS, hızölçer ve pusula donanımları kullanan mobil tarayıcı uygulamaları (Layar), ulusal alanda e-ticaret ve kurumsal web siteleri için AG uygulamaları (Basebros), AG ile endüstri, satış pazarlama, eğitim, perakende, sağlık ve ilaç, emlak konularında çözümler (Arimars), AG ile eş zamanlı uzaktan çalışma çözümleri (Digi-mode) şirketler bulunmaktadır.

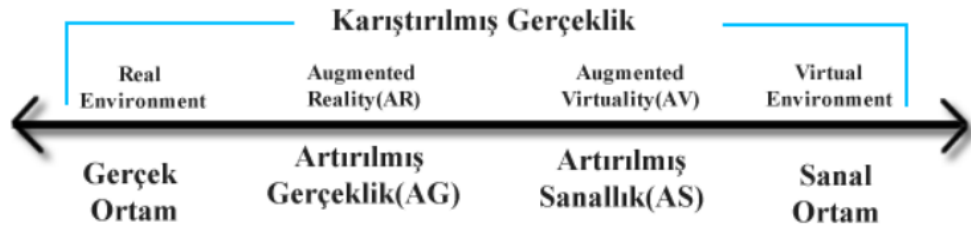
Literatürde işaretli AG uygulamalarında ses ile etkileşimin yapıldığı çalışmalara rastlanmaktadır. Bartie ve Mackaness (2006), Edinburgh şehrini ziyaret eden turistlere yardımcı olmak için belirli sesli komutları algılayan bir AG uygulaması geliştirmişlerdir. Mirzae vd. (2012), sesi girdi olarak kullanan işaretli AG uygulamasına bir örnektir. Kanno vd. (2018) ayrıca Alzheimer hastalarına yardımcı olmak için geliştirdikleri mobil AG uygulamasında sesli komutları yardımcı olarak kullandı. Sheldon vd. (2019) AG ortamında mimari tasarım modellerini uygulamak için ses tanıma ve el hareketi tanımayı kullanan bir yaklaşım önerdi. Siyayev ve Jo (2021) Boeing 737 uçağının bakım ve eğitimi için tasarladıkları AG uygulaması için bir ses tanıma sistemi önermişlerdir.

Bu çalışmada AG uygulamasındaki sanal nesnelere ses kontrolü sağlanmıştır. AG uygulaması Unity ortamında Vuforia AG paketi kullanılarak geliştirilmiştir. Ses tanıma modülünün gerçekleştirilmesi için Librosa Python kütüphanesinin özellik çıkarım yöntemleri kullanılmıştır. Librosa fonksiyonları ile elde edilen ses özellikleri, uzaklık ölçütü olarak DZB yönteminin kullanıldığı KEYK algoritmasına uygulanmıştır. Ayrıca ses özellikleri RO, KA ve DVM algoritmalarına uygulanarak karşılaştırmalı sonuçlar elde edilmiştir. Python ile geliştirilen ses tanıma modülü, Unity Python Scripting Paketi yardımıyla AG uygulamasına entegre edilmiştir.

Tezin geri kalan kısmı Őu Őekilde dŐzenlenmiŐtir: BŐlŐm 2’de AG, BŐlŐm 3’te literatŐrdeki AG uygulamaları ve BŐlŐm 4’te ses tanıma ile ilgili bilgiler verilmiŐtir. alıŐmada kullanılan materyal ve yŐntemler, donanım ile yazılım aralarının yanı sıra veri seti, Őzellik ıkarma yŐntemleri ve makine Őğrenmesi algoritmaları BŐlŐm 5’te aıklanmaktadır. Deneysel sonular BŐlŐm 6’da verilmiŐ ve BŐlŐm 7 alıŐmayı sonlandırmıŐtır.

## 2. ARTIRILMIŞ GERÇEKLİK

AG alanında tanım ve terimler, teknolojiye paralel olarak değişkenlik göstermektedir. Literatürde yer alan tanımlamalara bakıldığında, genel anlamda AG'nin bilgisayar ortamında oluşturulmuş 3B sanal nesnelere gerçek dünya görüntüleri ile bütünleşen bir teknoloji olarak tanımlandığı görülmektedir. Milgram vd. (1994)'nin sanal ve gerçek arasındaki ilişkiyi gösteren gerçeklik sanallık düzlemi Şekil 1'de gösterilmiştir.



Şekil 1. Gerçeklik sanallık düzlemi

AG uygulamaları, çalıştığı platform göz önüne alındığında optik tabanlı ve video tabanlı olarak iki temel kısma ayrılabilir (Azuma, 1997) . Optik tabanlı uygulamalarda, AG sahnesi özel bir gözlük ile algılanırken video tabanlı uygulamalarda bilgisayar, tablet, mobil telefon gibi bir cihazın ekranında görüntülenir (Somyürek, 2014).

### 2.1. AG İçin Kullanılan Donanım ve Yazılımlar

AG uygulamalarının geliştirilmesi ve kullanılması için bazı özel donanım ve yazılımlar gerekmektedir. Donanım alt yapısı için algılayıcılar (kamera, GPS, pusula, navigasyon, ivmeölçer, vb.) , ekranlar ve işlemci (bilgisayar, tablet ve mobil telefonlarının ekranları) gereklidir (Altınpulluk, 2015). Yazılım alt yapısı için ise ücretli ve ücretsiz alternatifler bulunmaktadır. Ücretli olanlara örnek olarak WikiTude ve LayAR, ücretsiz olanlara ise ARToolKit ve Vuforia SDK sayılabilir. Tablo 1'de AG uygulama geliştirme için kullanılacak yazılım araçları ve desteklenen platformlar verilmiştir. Bunların dışında 3B sanal nesnelere modellenmesini sağlayan Unity 3D, Blender, Google SketchUp gibi yazılım ortamları da mevcuttur. Modelleme yanında 3B nesnelere hareketini sağlayan motor yazılımını içeren Away3D, WebGL ve Unity3D uygulamaları, 3B nesnelere oluşturulmasını sağlayan 3DMax, Swift3D ve Unity3D uygulamaları bulunmaktadır.

**Tablo 1. AG Uygulama Geliştirme Yazılım Araçları**

AG Yazılım Araçları	Vuforia SDK	ARToolKit	WikiTude	LayAR
Desteklenen Platformlar	Windows, Android, IOS, Unity	Android, IOS, Windows, Linux, Mac, OS X, SGI	Android, IOS, Google Glass, Epson Moverio, Optinvent ORA1, PhoneGap, Xamarin	Android, IOS, BlackBerry

## 2.2. AG Türleri

AG türleri konum tabanlı ve görüntü tabanlı olmak üzere ikiye ayrılmaktadır (Cheng ve Tsai, 2013).

### 2.2.1. Konum Tabanlı Artırılmış Gerçeklik Uygulamaları (KTAG)

KTAG uygulamaları GPS, WiFi, RFID ya da sensör verileriyle kullanıcının konumunu belirleyerek, gerçek dünya görüntüleri üzerine sanal görüntü veya verileri ekleyerek AG ortamını zenginleştirirler (Sırakaya ve Seferoğlu, 2016). KTAG, eğitim, mimari, turizm, kültürel mirasın tanıtımı gibi alanlarda uygulamalar geliştirmek üzere kullanılabilir (İçten ve Bal, 2017). Wikitude AG uygulaması KTAG geliştirmek üzere sıkça kullanılan bir yazılım aracıdır (Kaleci vd., 2016). KTAG uygulamalarına çevre bilimi (Squire ve Klopfer, 2007), mobil navigasyon (Esengün, 2016), matematik (Dunleavy vd., 2009 ), doğal afetlerden korunma (Tsai vd., 2013) gibi alanlarda rastlanmaktadır.

### 2.2.2. Görüntü Tabanlı Artırılmış Gerçeklik Uygulamaları (GTAG)

AG uygulamalarının büyük çoğunluğu görüntü tabanlıdır (Abdüsselam ve Karal, 2012). GTAG ortamında resim, grafik, el hareketi, ses şeklinde tanımlanan nesnelere işaretçi olarak kullanılır ve işaretçinin algılanması ile yapılan analiz sonucunda belirli noktalara

sanal nesnelere, veriler veya grafikler otomatik olarak eklenmektedir. GTAG uygulamaları İT ve İşaretçi Tabanlı Olmayan (İTO) şeklinde ikiye ayrılmaktadır. Akademik alanda İTAG uygulamalarına, işaretçi tabanlı olmayan artırılmış gerçeklik (İTOAG) ve KTAG uygulamalarına göre daha çok rastlanmaktadır (İçten ve Bal, 2017).

#### 2.2.2.1. İşaretçi Tabanlı Artırılmış Gerçeklik Uygulamaları (İTAG)

Bu tür AG uygulamalarında görüntülenmesi beklenen sanal nesne ile ilişkilendirilecek işaretçinin sisteme önceden yüklenmesi veya tanıtılması gereklidir. Sanal nesne, hareketli kullanıcının yolu üzerindeki bir görüntü ya da kameranın yöneltildiği kitaptaki bir görüntü ile ilişkilendirilebilir. İTAG uygulamalarının geliştirilmesi diğer türlere göre daha kolaydır. Ayrıca bu tür uygulamaların geliştirilmesi için kullanılacak yazılım araçlarının çeşitliliği ve ek donanımlara ihtiyaç duyulmaması geniş uygulama alanına sahip olmalarını sağlamaktadır. İTAG aynı zamanda görüntü tanıma AG olarak da adlandırılır.

#### 2.2.2.2. İşaretçi Tabanlı Olmayan Artırılmış Gerçeklik Uygulamaları (İTOAG)

İşaretsiz takip uygulamaları olarak da adlandırılan İTOAG uygulamaları, işaretçi verisi yerine AG ortamındaki bir nesnenin hareketi veya duvar, el gibi bir yüzeye AG ortamının yansıtılması ile etkileşimde bulunarak çalışır. İlgili eylem el hareketi veya ses olabilir. Mirzae vd. (2012) tarafından yapılan ve duyma engelli kişilerin konuşmasını yazıya çeviren AG uygulaması bir İTOAG örneğidir. Bu tür AG uygulamalarının en önemli özelliği sanal nesnelere görüntüleneceği yerin sabit olmamasıdır (Sırakaya ve Seferoğlu, 2016). Örneğin; yol çizgileri, trafik işaretleri veya yaya geçidi çizgileri algılanarak uygun geribildirimlerin AG ortamında otomatik olarak alındığı uygulamalar geliştirilebilir. KTAG uygulamaları da aynı zamanda İTO uygulamalarıdır. Sanal nesnelere AG ortamında görüntülenmesi GPS, WiFi, RFID gibi veri kaynakları kullanılarak sağlanır.

#### 2.2.2.3. Projeksiyon Tabanlı Artırılmış Gerçeklik Uygulamaları (PTAG)

PTAG uygulamaları, ışığı bir yüzeye üzerine düşürmeyi sağlayan küçük bir projektör gerektirir. PTAG'a örnek olarak verilebilecek Google Glass, ışığı küçük bir prizmaya

düşüren küçük bir projektöre sahiptir. Bu sayede gerçek dünya görüntüsü üzerine zaman, hava durumu, konum gibi bilgilerin sanal biçimde eklenmesi mümkündür. Google Glass, işlemci, bellek, WiFi, bluetooth, kamera, ekran, mikrofon, hoparlör, ivmeölçer, jiroskop gibi pek çok donanım içermektedir.

#### 2.2.2.4. Süperpoze Tabanlı Artırılmış Gerçeklik Uygulamaları (STAG)

STAG uygulamalarında sanal nesnelere gerçek nesnelere üzerine bindirilir ve nesne izleme yöntemi ile gerçek nesnelerin yerlerini alırlar. Bu yüzden nesne izleme AG olarak da adlandırılır.

### 3. LİTERATÜRDEKİ AG UYGULAMALARI

Literatürdeki çalışmalarda çoğunlukla ARToolKit ve Vuforia SDK yazılım geliştirme araçları kullanılmıştır. Özellikle, Vuforia SDK mobil cihazlar üzerinde çalışabilmesi, basit 3B nesne oluşturabilmesi, bilgisayarla görme teknolojisine sahip olması ve taşınabilir cihazların farklı işletim sistemi sürümlerine destek verebilmesi gibi özelliklerinden dolayı akademisyenler tarafından tercih edildiği düşünülmektedir.

#### 3.1. Eğitim Alanındaki Çalışmalar

Eğitim alanında AG uygulaması olarak pek çok akademik çalışma yapılmıştır. İbili ve Şahin (2013) tarafından geliştirilen AG uygulaması geometrik şekillerin 3B sanal nesnelere olarak görüntülenmesini sağlamıştır. Ortaöğretim geometri kitabının ilgili sayfasında 2B olarak görünen geometrik şekillerin, kamerada görüntülenen 3B bir kopyasının üzerinde 3B olarak görüntülenmesi sağlanmıştır. Şekil 2’de ilgili AG uygulamasına ait bir ekran görüntüsü görülmektedir.



**Şekil 2. İbili ve Şahin (2013) tarafından geliştirilen AG uygulamasına ait bir ekran görüntüsü**

Abdusselam ve Karal (2012) ortaöğretim fizik dersine ait manyetik alan konusunu AG uygulaması olarak gerçekleştirmişlerdir. Ekranda sanal olarak görüntülenenen manyetik şekiller iki mıktanıs arasında oluşan manyetizma etkisini öğrencilere göstermeyi amaçlamaktadır. Perez-Lopez ve Contero (2013) insan vücudundaki dolaşım ve

sindirim sistemlerini AG uygulaması ile öğertilmesi için bir çalışma yapmışlardır. Uygulamada görüntülenen sanal düğmelere tıklandığında ilişkili organın 3B görüntüsü ile birlikte açıklayıcı bilgiler ekranda ortaya çıkmaktadır. Şekil 3'te ilgili AG uygulamasına ait ekran görülmektedir.



**Şekil 3. Perez-Lopez ve Contero (2013) tarafından geliştirilen AG uygulamasına ait bir ekran görüntüsü**

Di Serio vd. (2012) tarafından görsel sanatlar dersinin AG ortamında gerçekleştirilmesi ile ilgili çalışmalarında Rönesans dönemine ait bazı eserleri sanal ortamda AG teknolojileri ile video, metin, ses ve 3B modeller ekleyerek zenginleştirmişlerdir. Şekil 4'te uygulamaya ait bazı ekranlar verilmiştir.



**Şekil 4. Di Serio vd. (2012) tarafından geliştirilen AG uygulamasına ait ekran görüntüleri**

Juan vd. (2010) ilkökul öğrencilerinin kelime öğrenimi için bir AG oyun uygulaması geliştirmişlerdir. Otuz iki ilkökul öğrencisi hem AG oyununu hem de gerçek oyunu oynamışlardır. Yapılan karşılaştırmada öğrencilerin sanal ve gerçek oyun arasında bir



fark görmedikleri fakat %81'nin AG oyununu daha çok sevdikleri anlaşılmıştır. Şekil 5'te uygulamaya ait ekran görüntüleri verilmiştir.



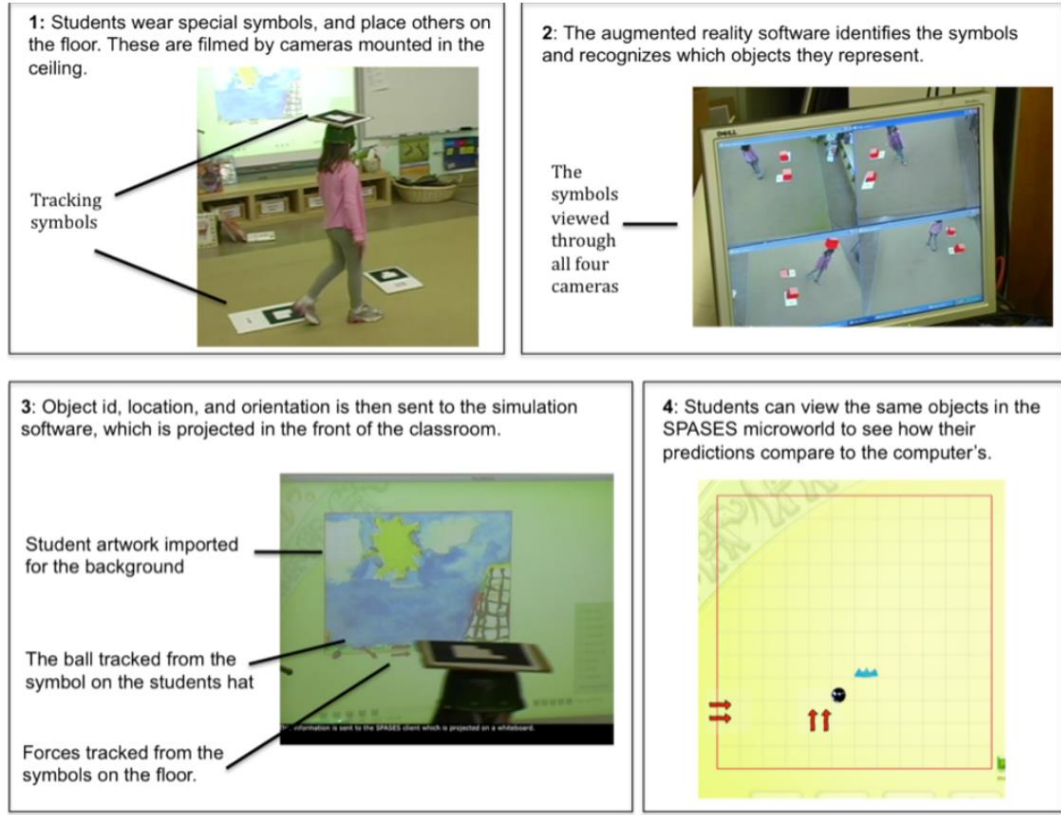
**Şekil 5. Juan vd. (2010) geliştirdiği AG uygulamasına ait bir ekran görüntüsü**

Erbaş ve Demirer (2014), eğitim alanında yapılmış ve Google Glass kullanılan AG uygulamalarını incelediği yayınlarında, Google Glass teknolojisi hakkında bilgi verdikten sonra eğitim alanında dünyada yapılmış AG uygulamaları hakkında detaylı bilgiler vermişlerdir.

Enyedy vd. (2016), fizik derslerinde uygulanmak üzere AG ortamında hareket yakalama teknolojisini içeren bir sistem önermişlerdir. Altı ila sekiz yaş arasındaki öğrencilere fizikteki kuvvet, hareket, hız, sürtünme konularının uygulamalı olarak öğretilmesi amaçlanmıştır. Sınıf tavanına yerleştirilen projeksiyon cihazı ile öğrencilerin hareketlerinin yakalabilmesi sağlanmıştır. Öğrenciler kendilerini ve tüm hareketlerinin etkilerini projeksiyon ekranında izleyebilmektedir. Ayrıca iki nesnenin birbiriyle etkileşmesi sonucu meydana gelen hareketleri gözlemleyebilmektedir. Çalışmanın sonucuna göre uygulamanın öğretici ve eğlendirici olduğu fakat öğrencilerin kendilerinden beklenen hareketleri yapmakta zorlandıkları ve projeksiyon ekranındaki bazı görüntüleri yorumlamakta güçlük çektikleri ifade edilmektedir. Şekil 6'da uygulamaya ait örnek görüntüler verilmiştir.

Yang ve Liao (2014), öğrencilerin dil ve kültür öğrenme süresinin kısaltılmasını sağlamak üzere Virtual English Classroom AR adını verdikleri bir İTOAG uygulaması geliştirmişlerdir. Belirli el hareket hareketlerine göre sanal nesnelerin görüntülenmesi, döndürülmesi ve yer değiştirmesi sağlanmaktadır. El açma kapama sanal nesnenin görüntülenmesi, iki parmak ucunun birbirine yaklaşıp uzaklaşması, boyut değişimi, tek

parmağın hareket ettirilmesi ise nesnenin yer değiştirmesi için kullanılmıştır. Şekil 7’de ilgili uygulamaya ait görüntüler verilmiştir.



Şekil 6. Enyedy vd. (2016) geliştirdiği AG uygulamasını açıklayan görüntüler



Şekil 7. Yang ve Liao (2014)'ya ait AG uygulamasını açıklayan görüntüler

Parmar vd. (2015), mühendislik öğrencilerine yönelik olarak teknik resim gibi grafik teknik bilgisinin öğretildiği dersler için AG uygulamaları geliştirmişlerdir. Böylece 3B nesnelerin farklı açılardan görüntülerinin 2B olarak kitaplarda resmedilmesinin algıyı zorlaştırmasının önüne geçmeyi amaçlamışlardır. Nesnelerin sanal biçimde 3B olarak AG ortamında görüntülenerek döndürme, öteleme gibi hareketler ile farklı yönlerinin görüntülenmesi sağlanmaktadır.

Zauniddin vd. (2010), işitme engelli öğrencilerin kullanabileceği bir AG uygulaması tasarlamışlardır. İlkokul beşinci sınıf fen bilgisi dersi için mikroskop uygulaması yapılmıştır. AG için metin, işaret dili videoları ve 3B sanal görüntü kullanılmıştır.

Eğitim alanında gerçekleştirilen AG uygulamalarının genel olarak öğrenme sürecini hızlandırdığı, motivasyon ve hoşnutluğu artırdığı, konuları daha anlaşılır hale getirdiği görülmüştür. Ayrıca öğrencilerin konuya dikkatlerinin kolaylıkla çekildiği, derse ilgisinin canlı tutulduğu belirlenmiştir. Öte yandan, zaman ve lokasyona bağlı olmadan öğretim olanağı ile birlikte, zihinde canlandırmanın zor olduğu durumları görselleştirme gibi avantajları sağlamaktadır.

### **3.2. Sanat Alanındaki Çalışmalar**

Coşkun (2017), AG teknolojisinin yeni sanat akımları ve sanat etkinliklerinde kullanıldığını belirtmiştir. AG kullanımı ile 2B görseller veya sergilerde yer alan sanat eserlerini hareketli hale getirmek, animasyon veya video ekleyerek ilgi çekici hale getirmek mümkündür. Öte yandan, AG ile oluşturulan sergiler fazla vakit alabilmekte ve dikkat dağıtıcı olabilmektedir. Çalışmada, AG teknolojisinin kullanıldığı farklı sanat etkinlikleri, müzeler ve sergilerde örnekler verilmiştir. Şekil 8’de bir müzede sergilenen eserlerle AG uygulaması üzerinden yaşanan etkileşim gösterilmiştir.



**Şekil 8. Bir müzede AG uygulaması ile yaşanan deneyim**

### **3.3. Trafik Alanındaki Çalışmalar**

Zhang (2016) trafik tıkanıklığını gidermek ve akışını kontrol etmek için gerçek zamanlı bir AG uygulaması geliştirmiştir. Gerçek zamanlı yöntemler geçmiş araç ve yaya yoğunluğu sayıları yerine anlık verileri dikkate almaktadır. Yapılan çalışmada Unity 3D

ile gerçek dünyada var olan binalar, yollar, ağaçlar, bulutlar gibi nesnelere ve bunların etkileşimini belirleyen betikler oluşturulmuştur. Uygulamaya eklenen düğmeler ile kullanıcı günün saatini ve hava durumunu değiştirebilmektedir. AG uygulaması buna göre trafikteki araç ve yayaların hareketini otomatik olarak düzenlemektedir. Şekil 9’da geliştirilen trafik AG uygulamasına ait ekran görüntüsü verilmiştir.



**Şekil 9. Trafikle ilgili AG uygulaması ekran görüntüsü**

### **3.4. Doğal Afetlerle İlgili Çalışmalar**

Tsai vd. (2013) tarafından geliştirilen KTAG uygulaması kırsal bir bölgede bulunan sığınakların isimleri, konumları, yönleri, mesafeleri ve sığınakların kontenjanları ile her birinde kaç kişilik boş yer kaldığı bilgilerini akıllı telefon ekranında görüntülemektedir. Uygulamanın amacı doğal afet, savaş, terör saldırısı, nükleer felaket gibi durumlarda kullanıcının en kısa sürede en yakın ve içinde boş yer olan sığınağa ulaşmasını sağlamaktır.

Literatürde yukarıda verilen çalışmaların yanı sıra, veri tabanındaki resimler ile kameradan alınan görüntüleri eşleştirip getirme özelliğine sahip KTAG uygulaması (Takacs vd., 2008), turistlerin ziyaret ettikleri mekanlara ait bilgilerin kafaya monteli AG donanımı ile algılanarak ekrana yazı, resim şeklinde görüntülenmesi (Reitmayr ve Schmalstieg, 2004), kullanıcıların sanal nesne ile etkileşiminin sağlandığı çoklu ortam e-öğrenme AG uygulaması (Liarokapis vd., 2002), yayalar için önemli konumlara ait bilgilerin GPS ve işaretçi kullanılarak bulunulan bölgenin genel görünümünün sunulduğu AG uygulaması (Kolbe, 2004) gibi çalışmalar da bulunmaktadır.

### 3.5. AG Uygulamalarında Sesin Kullanıldığı Çalışmalar

Bartie ve Mackaness (2006), Edinburgh şehrini ziyaret eden turistlere yardımcı olmak için belirli sesli komutları algılayan ve sesli olarak yanıt dönebilen bir AG uygulaması geliştirmişlerdir. Bu uygulamada, harita verileri ve GPS kullanılarak örneğin belli bir coğrafi konumdaki bina önüne gelen kullanıcı o bina hakkında detaylı bilgi isteyebilmektedir. Bu durumda sesli yanıt sistemi ilgili bilgiyi seslendirmektedir. Ayrıca kullanıcı ses tanıma sistemini aktive etme, kapatma; sesli yanıt sistemiyle komutlar hakkında bilgi alma gibi isteklerde bulunabilmektedir.

Mirzae vd. (2012), duyma engelli insanların kullanabileceği bir AG uygulaması geliştirmişlerdir. Bunun için hazır olarak bulunan Dragon Naturally Speaking, Dragon Dictate ve Microsoft Speech Recognition otomatik ses tanıma motorlarını ve Intel AVCSR otomatik görsel ses tanıma motorunu denemişlerdir. Sesli yanıt sistemi için ise AT&T ve Google yazıyı sese çevirme motorlarını kullanmışlardır. Ses tanıma motorları için farklı testlerle doğruluk oranları ve çalışma zamanlarını elde etmişlerdir.

Kanno vd. (2018), erken aşama Alzheimer hastalarına yardımcı olmak için geliştirdikleri mobil AG uygulamasında sesli komutları yardımcı olarak kullanmışlardır. Hastalar komutlar yardımıyla örneğin ilaçları için hatırlatma, resimlerdeki kişileri sorma gibi etkileşimlerde bulunabilmektedirler. Hasta her ilacı için ayrı sorgulama yapabilmektedir. Ayrıca bu hastalardan sorumlu kişiler uygulama yardımıyla kaybolmaları halinde uygulama ile etkileşime geçip hastayı bulabilmektedirler.

Sheldon vd. (2019) AG ortamında mimari tasarım modellerini uygulamak için ses tanıma ve el hareketi tanımayı kullanan bir yaklaşım önermişlerdir. Bu amaçla, 3B mimari modeller üzerinde görüntüleme ve değiştirme işlemleri yapabilmek için belli sayıda komutları ve bazı el hareketlerini tanıyan uygulama geliştirmişlerdir. Uygulama döndürme, büyütme, seçme, silme gibi sesli komutları kullanmaktadır. Sesli tanıma sistemi için Mac OS'un Speech Manager'ını kullanmışlardır.

Siyaeve ve Jo (2021) Boeing 737 uçağının bakım ve eğitimi için tasarladıkları AG uygulaması için bir ses tanıma sistemi önermişlerdir. Neuro-Symbolic Speech Executor (NSSE) olarak adlandırdıkları model, derin öğrenme ve sembolik akıl yürütme kullanarak sesli istekleri anlayıp uçaklarda bakım ve eğitim için gerekli belirli işlemleri yapabilmeyi sağlamaktadır. Bu model, uçak sistemlerine ait terimleri içeren karmaşık

sesli komutları anlayabilmektedir. Önerilen modelin, komutları %94.7 doğrulukla tanıdığını ve %7.5 kelime hata oranına sahip olduğunu belirtmişlerdir. Geleceğe yönelik çalışma olarak farklı doğal dillerdeki komutları da destekleyeceklerini ifade etmişlerdir.

#### 4. SES TANIMA

Ses tanıma sistemleri; sinyal işleme, akustik, model tanıma, haberleşme ve bilgi teorisi, dilbilim, psikoloji ve bilgisayar bilimi gibi birçok disiplini içinde barındırmaktadır (Aydın, 2005).

Ses tanıma sürecinde ilk olarak konuşmacının söyledikleri sisteme kaydedilir. Kaydedilen ses, işlenmemiş konuşma verisidir. İkinci olarak ses sinyali sayısallaştırılır. Sayısallaştırılan ses sinyali üzerinde sinyal analizi yapılarak sesin ayırt edici özellikleri ortaya çıkarılır. Daha sonra sinyal analizi sonucunda oluşan özellik vektörleri modellenir. Modellemenin sonucunda sözcüğün söylenişine karşılık gelen özellik vektörleri dizisinden bir sözcük modeli elde edilir ve bu model veri tabanında bulunan şablonlarla karşılaştırılır. Veri tabanında arama ve verilen sinyal için en iyi eşleşmeyi seçme ile ses tanıma işlemi gerçekleştirilir.

Bir ses tanıma sisteminin faydaları arasında kullanım kolaylığı sağlaması ve hızlı etkileşim vermesi sayılabilir. Mikrofon kullanıldığında, insanın alışkın olduğu özel çaba gerektirmeyen konuşma yöntemi veri giriş aracı olacağından kolaylık sağlar. Klavye, fare ve dokunmatik ekran gibi veri giriş araçlarına gerek kalmaz. Ayrıca bu araçlardan daha hızlı etkileşim sağlar.

Ses tanıma sistemlerinin kullanım alanları sesli ifadeleri metne çevirme, ses komutlarıyla kontrol, telefonla hizmet, engelli bireylere destek ve gömülü yazılımlar şeklinde sıralanabilir.

Ses tanıma sistemlerinden farklı ihtiyaçlar veya amaçlarla yararlanan kullanıcılar 3 grupta toplanabilir.

1. Nörolojik veya patolojik nedenlerle ya da kazalara bağlı olarak ellerini kullanamayan insanlar.
2. Yazı yazmak için yeterli imkanı, zamanı olmayanlar veya iyi yazamayan profesyonel insanlar. Doktorlar, avukatlar, psikologlar, satış elemanları gibi çalışanlar örnek olarak verilebilir.
3. Doğru yazmayı öğrenmelerini engelleyen öğrenme zorluğu çeken insanlar.

Ses tanıma sistemlerini aşağıdaki açılardan sınıflandırmak mümkündür:

1. Konuşmanın sürekliliğine göre:

Ayrık kelimelerin tanınması yapılan sistemlerde, kelimeler arasında boşluk olması beklenir. Dikteyle yazdırma yapılan sistemlerde konuşmacı doğal biçimde konuşur. Bağlı kelimeleri tanıma uygulamalarında kelimeler arası boşluklar daha kısadır ve birden fazla kelimenin tanınması hedeflenir. Konuşma tanıma uygulamalarında ise, insanın doğal biçimde konuşması esnasında çıkardığı ara seslerin de elimine edilmesini sağlayacak yöntemler uygulanarak ses tanıma gerçekleştirilmeye çalışılır.

2. Konuşmacıya bağlı olup olmamasına göre:

Konuşmacıdan bağımsız ses tanıma yapabilmek için veri kümesinin farklı kişilere ait sesleri de içererek öğrenme algoritmalarının eğitilmesi gereklidir. Veri kümesi tek kişiye ait seslerden oluşursa doğal olarak farklı kişilerin seslerini tanımakta başarısız olma olasılığı artacaktır. Ses veri kümesindeki kayıtlar aynı kişiye ait olsa bile kişinin ruhsal durumu, hastalıklar gibi nedenlerle sinyal özellikleri farklılık gösterebilmektedir. Dolayısıyla, konuşmacıdan bağımsız ses tanıma sistemlerinin uygulanması daha da zordur (Baygün, 2006).

Ses tanımayı zorlaştıran nedenlerin en başında ses sinyalinin ortam gürültüsünden kolaylıkla etkilenmesi gelmektedir. Bu sorunu gidermek için kullanılan en yaygın yöntem filtre kullanımıdır. Diğer bir sorun ise, aynı komuta ait ses sinyalinin kişiden kişiye, hatta aynı kişinin farklı zamanlarda seslendirilişine göre bile değişik nedenlerle değişebilmesidir. Yine benzer olarak, şive ve aksana bağlı değişiklikler de söz konusudur.

#### **4.1. Ses Tanıma Adımları**

Bu çalışmadaki ses tanıma uygulamasına ait adımlar eğitim ve test olmak üzere iki kısımda gruplandırılabilir.

Eğitim aşamasına ait adımlar:

1. Ses sinyallerinin kaydedilmesi
2. Ses sinyalleri içindeki ifadenin saptanması



3. Gürültünün giderilmesi için sinyallere filtre uygulanması
4. Sinyal özelliklerinin çıkarılması
5. Çıkarılan özellikler ile makine öğrenmesi algoritmasının eğitilmesi

Test aşamasına ait adımlar:

1. Ses sinyalinin alınması
2. Ses sinyali içindeki ifadenin saptanması
3. Gürültünün giderilmesi
4. Sinyal özelliklerini çıkarılması
5. Eğitilmiş olan makine öğrenmesi algoritmasına özelliklerin verilerek tahmin sonucunun alınması

## 5. MATERYAL VE YÖNTEMLER

### 5.1. Yazılım Paketleri

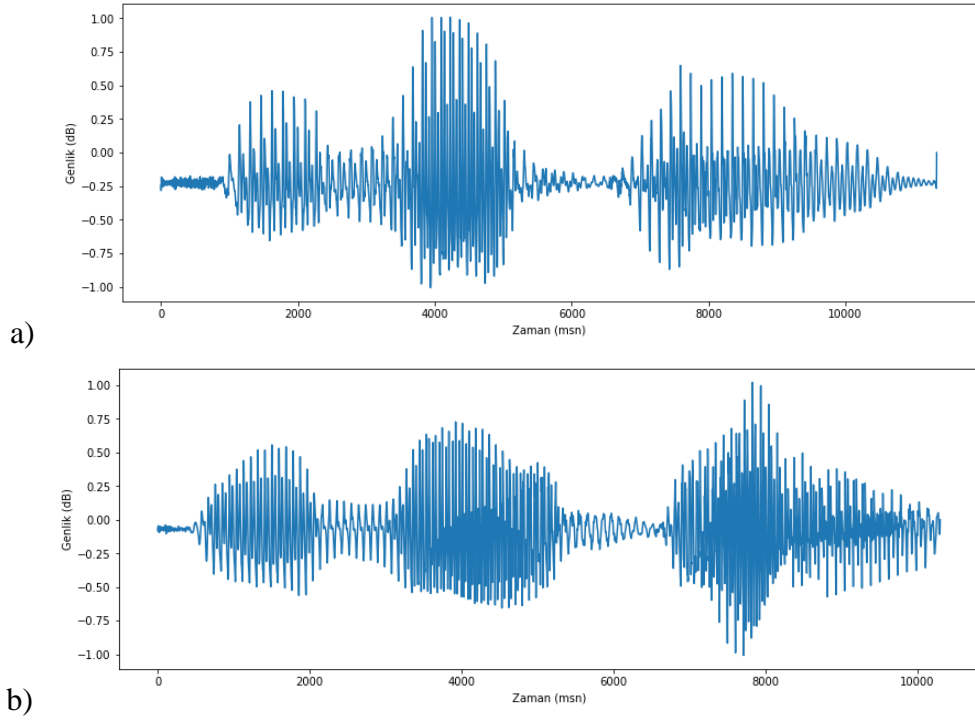
Bu çalışmada, AG uygulamasının oluşturulması için Vuforia 10.3.2 AR Engine ile Unity 2020.3.3f1 kullanılmıştır. Ses tanıma Python modülümüzün Unity'ye entegrasyonu için Unity Python Scripting 6.0.0 paketi kullanılmıştır. Bu paket, Python Çalışma Zamanı Ortamını bir ad alanı olarak içe aktarma ve Unity Düzenleyicisi içinde Python betiklerini çalıştırma araçlarını sağlar. Ses özellik çıkarımı için Librosa 0.9.2 Python kütüphanesi kullanılmıştır. Ses tanıma modülünün uygulanması için Python 3.7 ile Spyder IDE kullanılmıştır. İşletim Sistemi Windows 11 Home Edition idi.

### 5.2. Donanımlar

Bu çalışma sırasında ses kayıtları için 6.0.8921.1 sürümlü Realtek(R) Audio Microphone Array kullanılmıştır. Giriş formatı 2 kanal, 16 bit ve 48000 Hz idi. Bilgisayar sistemi, 16 GB RAM ve 2.60 GHz Intel Core i7-9750H CPU'ya sahip Lenovo Legion Y540-15IRH Dizüstü bilgisayardı. Grafik kartı NVIDIA GeForce RTX 2060 idi.

### 5.3. Kullanılan Veri Kümesi

Veri seti, farklı cinsiyetten iki kişiden alınan beş farklı Türkçe komutun ses kayıtları ile oluşturulmuştur. Kaydedilen sesli komutlar “Dur”, “İlerle”, “Sola Dön”, “Sağa Dön” ve “Geri Gel” dir. Ses kaydı hem gürültülü hem de gürültüsüz ortamlarda gerçekleştirilmiştir. İlk olarak, 5 komutun her biri 10 kez kaydedilmiş ve her insan için 50 kayıt verilmiştir. Dolayısıyla veri seti toplamda 100 ses kaydı içermektedir. Kayıt gerçekleştirildikten sonra seslerin sessizlik kısımları kaldırılmıştır. Örnek olarak Şekil 10'da her iki cinsiyetten (sırasıyla kadın ve erkek) alınan “Sola Dön” komutunun ses kayıtlarının grafiği görülmektedir. Makine öğrenmesi algoritmalarının performansını değerlendirmek için ses kayıtlarından MFKK, Melspectogram ve DTK öznitelikleri çıkarılmıştır. 100 ses kaydından oluşan ilk veri seti ile makine öğrenimi algoritmalarının performanslarını değerlendirdikten sonra, daha fazla analiz yapmak için veri seti 100 ses daha ile genişletilmiştir. Bu nedenle, genişletilmiş veri setinin toplam boyutu 200 ses kaydını içermektedir.



**Şekil 10. “Sola Dön” komutu ses kaydı a) Kadın b) Erkek**

#### 5.4. Ses Kayıtlarından Özellik Çıkarma

Aşağıdaki özellikler ses kayıtlarından çıkarılmış ve makine öğrenimi algoritmalarına uygulanmıştır.

##### 5.4.1. Mel-Frekans Kepstral Katsayıları (MFKK)

MFKK ilk olarak Davis ve Mermelstein (1980) tarafından tanımlanmıştır. Konuşma tanımada MFKK'nin birçok başarılı uygulaması vardır. Mel-ölçekli filtre bankası, insan kulağının frekans seçiciliğini taklit ederek iyi bir konuşmacı ayrımı sağlar. Ayrıca MFKK katsayıları, ses dalgası yapısındaki değişikliklerden çok daha az etkilenir. MFKK öznelik çıkarma yönteminin detayları aşağıda tanımlanmıştır.

Fast Fourier Transform (FFT) hesaplamasının tüm sinyaldeki frekans değişikliklerinden etkilenmemesini sağlamak için sinyal önce kısa çerçevelere bölünür. Hamming pencereleme yöntemi ile çerçevelerin sonunda oluşacak spektral değişimlerin önüne geçilmektedir. Her çerçeve  $n$  için Denklem 1 uygulanır.

$$w_n = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (1)$$

Burada,  $0 \leq n \leq N-1$  ve  $N$ , pencere uzunluğudur. Ardından, her çerçeve için FFT hesaplandıktan sonra güç spektrumu Denklem 2'deki gibi hesaplanır.

$$P = \frac{|FFT(x_i)|^2}{N} \quad (2)$$

Burada  $x_i$ , sinyalin  $i$ 'inci çerçevesidir.

Bir sonraki adımda, mel ölçeğinde üçgen filtreler uygulanarak filtre bankaları hesaplanır. Frekans  $f$  için mel değeri, Denklem 3'teki gibi hesaplanır.

$$mel(f) = 2595 \times \log_{10}\left(1 + \frac{f}{700}\right) \quad (3)$$

Filtre bankası katsayıları yüksek oranda ilişkili olduğundan, sıkıştırma, Ayrık Kosinüs Dönüşümü (AKD) uygulanarak elde edilir. Ses tanıma uygulamaları tipik olarak 2-13 MFKK özelliğini kullanır.

#### 5.4.2. Doğrusal Tahmini Kodlama (DTK)

DTK'deki temel fikir, önceki ses örneklerinin doğrusal bir kombinasyonunu kullanarak bir ses örneğini tahmin etmektir (Kurzekar ve vd., 2014).

DTK yönteminin ilk iki adımı, MFKK yönteminde olduğu gibi çerçeveleme ve pencerelemedir. Bir sonraki adım, pencereli sinyalin her karesini otomatik olarak ilişkilendirmektir. En yüksek otokorelasyon değeri, DTK kodlayıcının sırasındadır. Bir sonraki adımda, otokorelasyon çerçevelerinden DTK cepstral katsayıları elde edilir.

Tahmin hatası, tahmin edilen değer ile gerçek değer arasındaki farktır. Mevcut numune  $x_i$ , önceki  $p$  numuneler tarafından tahmin edilecekse ve  $x_i^p$  tahmin edilen değer ise, o zaman:

$$x_i^p = -a_2x_{i-1} - a_3x_{i-2} - \dots - a_{p+1}x_{i-p} \quad (4)$$

Burada  $\{1, a_2, \dots, a_{p+1}\}$   $(p+1)$  adet filtre katsayılarıdır. Sinyal, bir özellik vektörüyle sonuçlanan bir DTK filtresinden geçer.

#### 5.4.3. Mel-spectrogram

Mel-spektrogram özelliklerini elde etmek için önce spektrogram hesaplanır. Bunun için FFT, sinyalde üst üste binen pencereler üzerinde hesaplanır. Ardından frekans, spektrogramla sonuçlanan  $\log$  ölçeğine dönüştürülür. Son olarak, mel-spektrogramını elde etmek için frekans Denklem 3'te belirtildiği gibi mel ölçeğine eşlenir.

## 5.5. Makine Öğrenmesi Algoritmaları

Ses kayıtlarından çıkarılan özellikler, aşağıdaki makine öğrenimi algoritmalarına uygulanmıştır.

### 5.5.1. Karar Ağacı (KA)

Bir veri kümesinden KA oluşturan algoritma ilk olarak Quinlan (1993) tarafından ID3 (Iterative Dichotomiser 3) olarak tanıtılmıştır. C4.5 algoritması, hem ayrık hem de sürekli değerleri işlemek için ID3 algoritmasının bir uzantısıdır. C4.5 aynı zamanda eksik değerleri ve nitelikleri farklı maliyetlerle ele almaktadır. C4.5, bilgi teorisinden entropi kavramını Denklem 5 kullanarak ağaç yapısını oluşturur.

$$Entropy(S) = - \sum_{i=1}^c p_i \log_2(p_i) \quad (5)$$

Algoritma, ağacı oluştururken en yüksek bilgi kazancına Denklem 6 sahip bir sonraki özneliği seçer.

$$Gain(S, A) = Entropy(S) \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (6)$$

C4.5 algoritması ayrıca oluşturulduktan sonra KA'yı basitleştirmek için budama uygular.

### 5.5.2. Rastgele Orman (RO)

RO algoritması küçük karar ağaçları oluşturur ve bunların çıktılarını torbalamaya dayalı topluluk mantığını kullanarak birleştirir (Breiman, 2001). Her küçük KA zayıf bir sınıflandırıcıdır. Orman, veri kümesinin boyutuna ve özellik sayısına göre değişen çok sayıda küçük ağaçtan oluşur. Küçük ağaçların oluşturulması sırasında, düğümlerde bir sonraki özelliği seçmek için genellikle Gini Endeksi kriteri Denklem 7'deki gibi kullanılır.

$$Gini = 1 - \sum_{i=1}^c (p_i)^2 \quad (7)$$

İlk olarak, bootstrapping kullanılarak, veri kümesindeki satırlar değiştirilip rastgele seçilerek küçük eğitim kümeleri oluşturulur. Bu durumda, orijinal veri kümesinin bazı satırları hiç seçilmeyebilir. Bu satırlar, RO'nun torba dışı hatasını ölçmek için kullanılır. Ayrıca ağaçlardaki düğümler bölünürken tüm özelliklerin yerine rastgele seçilen birkaç özellik değiştirilmeden kullanılır.

### 5.5.3. Destek Vektör Makineleri (DVM)

DVM'nin genel fikri, farklı sınıflardaki eğitim örneklerini, örneklerin alındığı alanda marjı olan bir hiper düzlem kullanarak ayırmaktır. DVM algoritması, bu marjı maksimize ederken sınıflandırma hatasını da minimize etmelidir (Cortes ve Vapnik, 1995).

DVM'ler, girdi veri setini doğrusal olmayan bir eşleme işlevi kullanarak daha yüksek boyutlu bir özellik alanına eşler. DVM tarafından kullanılan tahmin fonksiyonu: Denklem 8'deki gibidir.

$$f(x) = (w \times \phi(x)) + b \quad (8)$$

Burada,  $w$  ve  $b$ , veri setinden tahmin edilen katsayılarıdır ve  $\phi(x)$ , özellik uzayındaki doğrusal olmayan fonksiyondur. Minimize edilecek risk fonksiyonu: Denklem 9'daki gibidir.

$$R(w, \xi^*) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (9)$$

$$d_i - w\phi(x_i) - b \leq \epsilon + \xi_i \quad (10)$$

$$(w\phi(x_i)) + b - d_i \leq \epsilon + \xi_i^* \quad (11)$$

Burada,  $C$  ceza gücüdür ve  $\xi_i, \xi_i^* > 0$ .

Destek vektörü tahmin fonksiyonu: Denklem 12'de verilmiştir.

$$f(x) = \sum_{i=1}^{NSV} (\alpha_i - \alpha_i^*) K(X, X_i) + b \quad (12)$$

Burada  $\alpha_i$  ve  $\alpha_i^*$  Lagrange katsayılarıdır ve  $NSV$  destek vektörlerinin sayısıdır.

Çekirdek fonksiyonu  $K(X, X_i) = \phi(X_i)\phi(X_j)$  polinom, radyal tabanlı fonksiyon (rbf) veya sigmoid olarak seçilebilir.

$C, \epsilon$  parametreleri ve kernel fonksiyonu kullanıcı tarafından belirlenir. Bu çalışmada kernel fonksiyonu olarak rbf (radial basis function) kullanılmıştır.  $C$  ve  $\epsilon$  için sırasıyla 1.0 ve 0.001 kullanılmıştır.

#### 5.5.4. k-En Yakın Komşu (kEYK)

kEYK algoritması, örneğe dayalı bir öğrenme yöntemidir. Tüm veri örnekleri, algoritmanın eğitim aşamasında saklanır. Ardından, kEYK, Denklem 13'teki gibi en yakın k veri örneğini kullanarak yeni bir örneği sınıflandırmak için belirli bir uzaklık ölçütü kullanır.

$$f(y) \leftarrow \operatorname{argmax}_{v \in V} \sum_{i=1}^k \delta(v, f(x_i)) \quad (13)$$

Burada  $\delta$  uzaklık fonksiyonu,  $v$  veri vektörü,  $V$  veri seti ve  $k$  komşu sayısıdır (Mitchell, 1997).

Bu çalışmada, yeni ses kayıtlarının sınıfını belirlemede uzaklık ölçümü için DZB yöntemi kullanılmıştır.

##### 5.5.4.1. Dinamik Zaman Bükme (DZB)

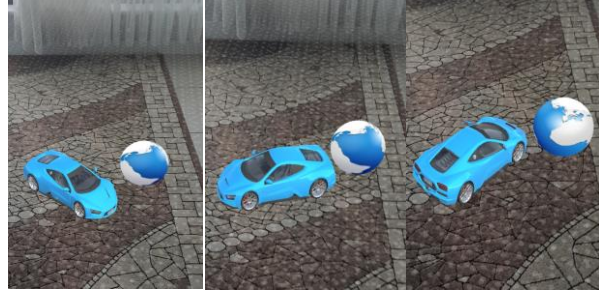
DZB yöntemi, zaman veya hız bakımından farklılık gösteren iki sinyalin benzerliğini ölçmek için kullanılır (Salvador ve Chan, 2004).

$P = p_1, p_2, \dots, p_n$  sinyallerinin benzerliği ve  $R = r_1, r_2, \dots, r_m$  bulunabilmek için önce  $n \times m$ 'lik bir matris oluşturulur. Bu matrisin  $i$ . ve  $j$ . öğeleri,  $r_i$  ve  $p_j$  noktaları arasındaki genlikteki uzaklık olan  $d(p_i, r_j)$ , döndürür. Tipik uzaklık ölçütü Öklid uzaklığıdır ve Denklem 14' verilmiştir.

$$d(p_i, r_j) = \sqrt{|p_i^2 - r_j^2|} \quad (14)$$

## 6. DENEYSEL SONUÇLAR

Bu çalışma iki farklı bölümden oluşmaktadır. Öncelikle Unity Editor içerisinde Vuforia AR Engine kullanılarak bir AG ortamı oluşturulmuştur. Şekil 11’de cep telefonu ekranındaki AG ortamı gösterilmektedir. Uygulama, sanal nesnelerin yüzeyde görüntülenmesi için işaretlidir.



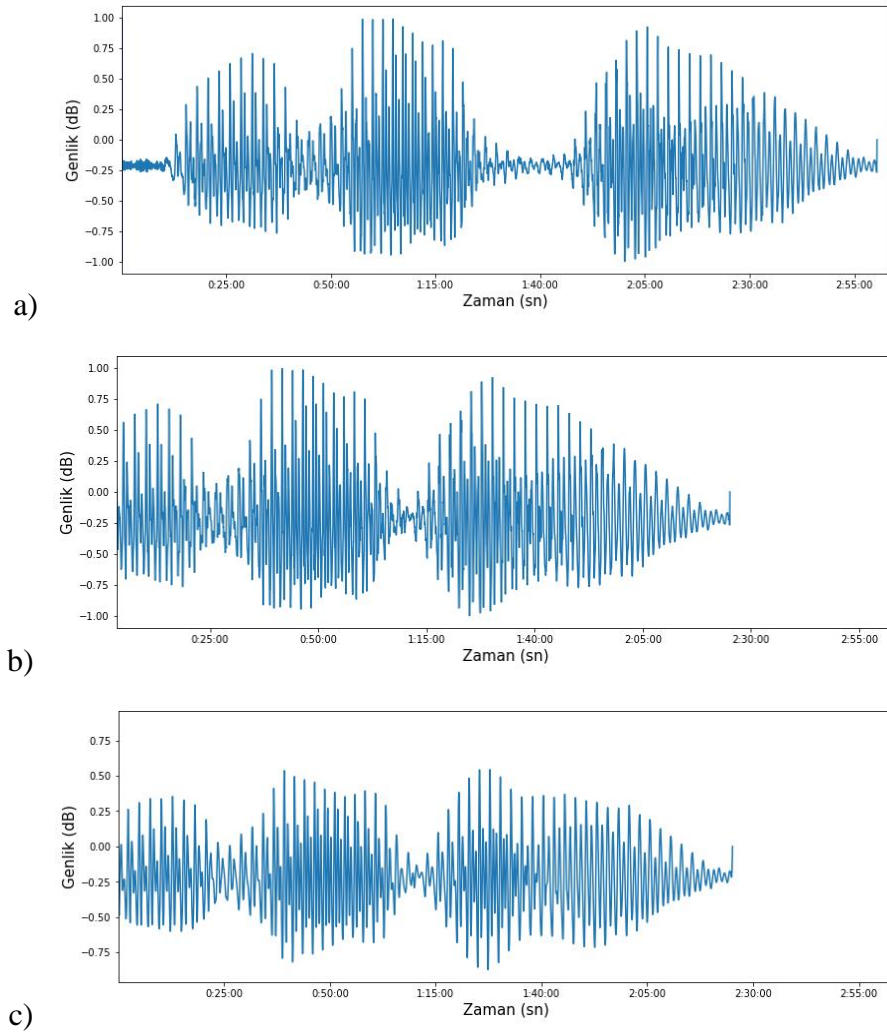
**Şekil 11. Bu tez çalışması için geliştirilen AG uygulaması**

Ardından, Spyder IDE içinde Python kullanılarak ses tanıma işlevi geliştirilmiştir. Python kodu, makine öğrenimi algoritmalarına uygulamak için Librosa (Brian vd., 2015) paketinin ses özellik çıkarma yöntemlerini ve scikit-learn kitaplık paketini kullanmaktadır.

Ses tanıma için ilk olarak ses veri seti oluşturulmuştur. Hem gürültülü hem de sessiz ortamlarda farklı cinsiyetten iki kişi tarafından beş farklı sesli komut kaydedilmiştir. Kayıtların her biri 8000 çerçeve boyutunda (fs) 2 saniye süreyle yapılmıştır. Bu nedenle arabellek başına çerçeve sayısı 16000’dir. Ses kayıtlarını yapmak için pyaudio Python kütüphanesi kullanılmıştır. Her kayıt yapıldıktan sonra pydub Python kütüphanesinin split\_on\_silence fonksiyonu kullanılarak sesteki sessiz kısım giderilmiştir. Python split\_on\_silence işlevi, sessiz bölümü algılamak ve kaldırmak için 100 ms olarak min\_silence\_length ve dBFS-1 olarak Silence\_threshold kullanarak sesi parçalara ayırmıştır. Sessiz olmayan kısımlar, AudioSegment sınıfının dışa aktarma işleviyle birleştirilmiştir. Ses kaydından gürültü giderme işlemi, Finite Impulse Response (FIR) filtresini iki kez uygulayan scipy.signal Python kitaplığının filtfilt işlevi kullanılarak yapılmıştır. Şekil 12, a) "Sola Dön" ses sinyalinin orijinal grafiğini, b) sessizliğin kaldırılmasından sonraki grafiği ve c) gürültü filtrelemeden sonraki grafiği göstermektedir.

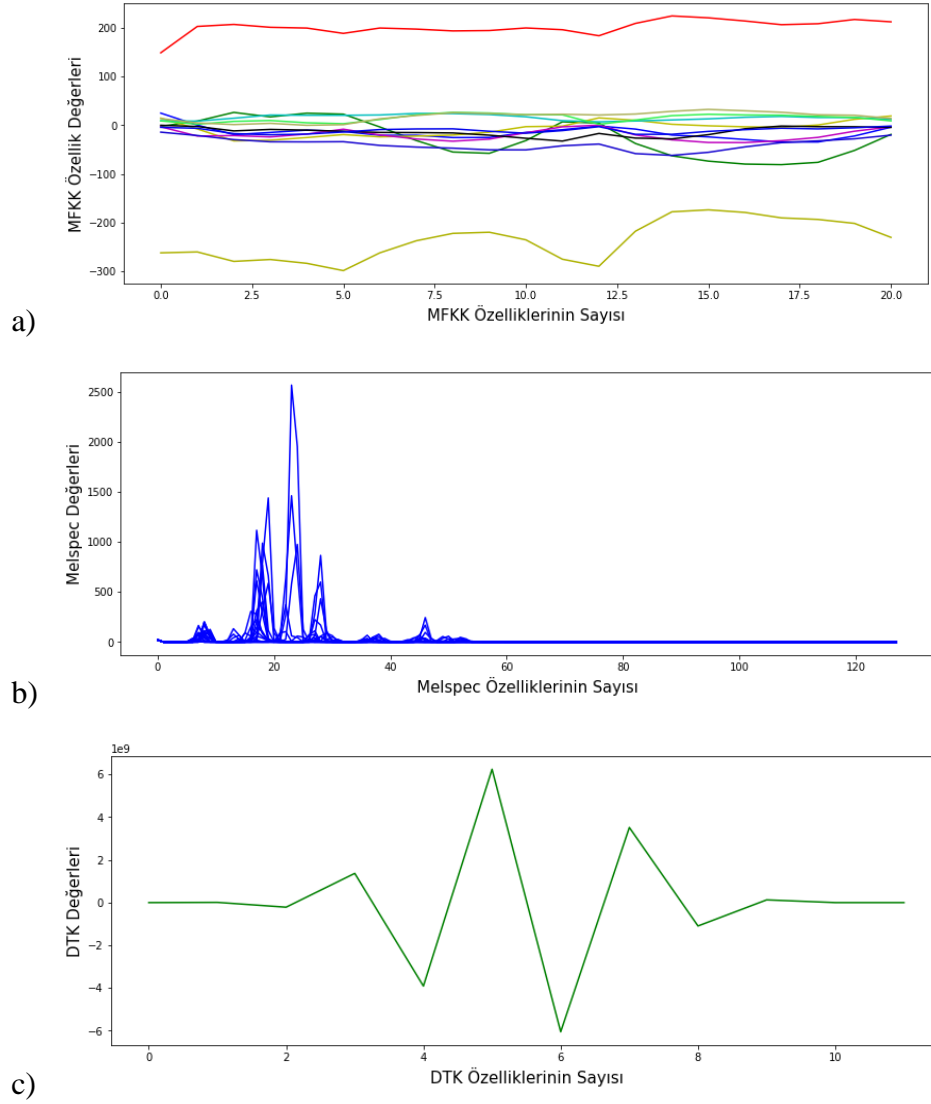


Sessizlik giderme ve gürültü filtreleme uygulandıktan sonra, Librosa Python kütüphanesinin MFKK fonksiyonu kullanılarak sinyallerden MFKK öznelikleri çıkarılmıştır. Çıkarılan MFKK özneliklerinin sayısı 12 ve atlama uzunluğu 512'dir. Şekil 12'de, orijinal ses sinyaline uygulanan sessizlik giderme ve gürültü filtrelemenin etkileri gözlemlenebilir. Şekil 13, sessizlik giderme ve gürültü filtrelemeden sonra "Sola Dön" ses sinyalinden çıkarılan MFKK değerlerini, Melspec değerlerini ve DTK değerlerini göstermektedir.



**Şekil 12. a) "Sola Dön" ses sinyalinin orijinal grafiği, b) Sessizliğin kaldırılmasından sonraki grafiği c) Gürültü filtrelemeden sonraki grafiği**

Sinyaller, MFKK özellikleri arasındaki uzaklık ölçütü olarak DZB kullanılarak KEYK algoritması ile sınıflandırılmıştır. DZB değerlerini hesaplamak için fastdtw Python işlevi kullanılmıştır. KEYK'deki komşu sayısı  $k$ , en iyi doğruluk oranını veren 6 olarak belirlenmiştir.



**Şekil 13. a) MFCC değerleri b) Melspec değerleri c) DTK değerleri**

Doğruluk oranları erkek ve kadın sesi veri setleri için sırasıyla %88 ve %92 olarak bulunmuştur. İki veri setinin birleştirilmesiyle oluşan veri setinin doğruluk oranı %84 olarak bulunmuştur. Erkek veri seti için KEYK'nin karışıklık matrisi Tablo 2'de verilmiştir.

**Tablo 2. KEYK algoritmasının erkek ses veri seti için karmaşıklık matrisi**

Gerçek \ Tahmin	Dur	İlerle	Sola Dön	Sağa Dön	Geri Gel
Dur	10	4			
İlerle		6			
Sola Dön			8		
Sağa Dön			2	10	
Geri Gel					10

Tablo 2'ye göre, İlerle sesli komutlarının 4'ü yanlışlıkla Dur olarak ve Sola Dön sesli komutlarından 2'si yanlış şekilde Sağa Dön olarak sınıflandırılmıştır.

Kadın veri seti için kEYK sınıflandırma sonucunun karışıklık matrisi Tablo 3'te verilmiştir.

**Tablo 3. kEYK algoritmasının kadın ses veri seti için karışıklık matrisi**

Gerçek Tahmin \	Dur	İlerle	Sola Dön	Sağa Dön	Geri Gel
Dur	10	1	1		
İlerle		7			
Sola Dön		2	9		
Sağa Dön				10	
Geri Gel					10

Tablo 3'e göre, İlerle sesli komutlarından 1 tanesi yanlışlıkla Dur ve 2 tanesi Sola Dön olarak sınıflandırılmıştır. Sola Dön sesli komutlarından bir tanesi Dur şeklinde yanlış sınıflandırılmıştır.

Birleştirilmiş veri kümesi için kEYK'nin karışıklık matrisi Tablo 4'te verilmiştir.

**Tablo 4. kEYK algoritmasının birleştirilmiş veri seti için karışıklık matrisi**

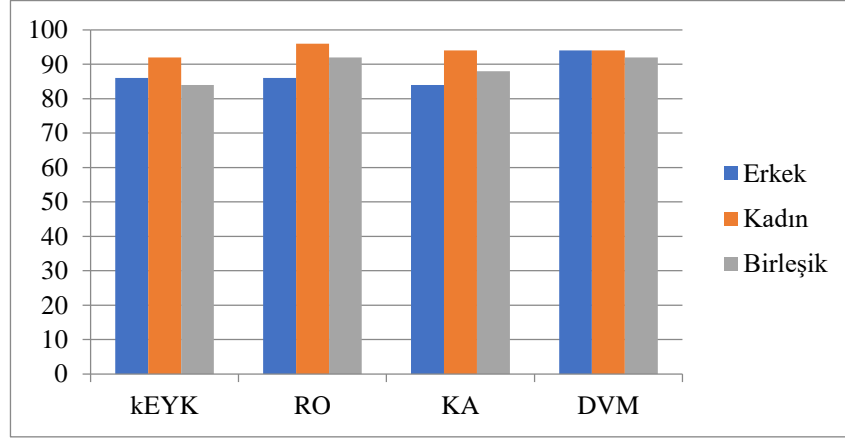
Gerçek Tahmin \	Dur	İlerle	Sola Dön	Sağa Dön	Geri Gel
Dur	20	3	1	2	
İlerle		12	4		
Sola Dön		4	13		
Sağa Dön		1	2	18	
Geri Gel					20

Tablo 4'e göre Dur ve Geri komutlarında herhangi bir yanlış sınıflandırma yapılmazken Sağa Dön komutunda sadece 2 yanlış sınıflandırma yapılmıştır. İlerle ve Sola Dön komutları, yalnızca erkek ve kadın veri kümelerinde olduğu gibi daha fazla yanlış sınıflandırmaya sahiptir.

Karşılaştırma amacıyla, sessizlik giderme ve filtreleme uygulandıktan sonra ses sinyallerinden MFKK, DTK ve Mel Spektrogram özellikleri çıkarılmıştır. Daha sonra bu özelliklerin ortalamaları ve standart sapmaları kullanılarak mfcc\_avg, mfcc\_std, lpc\_avg, lpc\_std, mel\_std ve mel\_avg olmak üzere 6 özellikle yeni veri setleri elde edilmiştir. Yeni veri setleri, sklearn.preprocessing kütüphanesinin MinMaxScaler fonksiyonu ile özellikler 0-1 aralığına ölçeklendirildikten sonra RO, KA ve DVM

makine öğrenimi (MÖ) algoritmalarına uygulanmıştır. Doğruluk oranlarını elde etmek için RO, KA ve DVM algoritmaları 3 kat çapraz doğrulama ile değerlendirilmiştir.

Ortalama doğruluk oranları Şekil 14'te verilmiştir.



**Şekil 14. İlk veri seti üzerinde algoritmaların ortalama doğruluk oranları**

Şekil 14'e göre DVM algoritması erkek veri setinde %94 ile en yüksek doğruluğa sahipken, RO kadın veri setinde %96 ile en yüksek doğruluğu vermiştir. RO ve DVM, birleşik veri seti için %92 ile en yüksek doğruluğa sahiptir.

Birleşik veri seti için DVM sınıflandırma sonucunun karışıklık matrisi Tablo 5'te verilmiştir.

**Tablo 5. Birleşik veri kümesi için DVM algoritmasının karmaşıklık matrisi**

Gerçek \ Tahmin	Dur	İlerle	Sola Dön	Sağa Dön	Geri Gel
Dur	18	1			
İlerle	2	19	2		
Sola Dön			18	3	
Sağa Dön				17	
Geri Gel					20

Tablo 5'e göre, Sağa Dön komutlarından 3'ü yanlışlıkla Sola Dön, Sola Dön komutlarından 2'si yanlışlıkla İlerle olarak sınıflandırılmıştır. Dur komutlarından 2 tanesi İlerle olarak yanlış sınıflandırılırken ve İlerle komutlarından 1 tanesi, Dur olarak yanlış sınıflandırılmıştır.

Birleşik veri seti için RO sınıflandırma sonucunun karışıklık matrisi Tablo 6'da verilmiştir.

**Tablo 6. Birleşik veri kümesi için RO algoritmasının karmaşıklık matrisi**

Gerçek \ Tahmin	Dur	İlerle	Sola Dön	Sağa Dön	Geri Gel
Dur	18	1	1		
İlerle	2	19			1
Sola Dön			18	2	
Sağa Dön			1	18	
Geri Gel					19

Tablo 6'ya göre, Sağa Dön komutlarından 2'si yanlışlıkla Sola Dön, Sola Dön komutlarından 1'i yanlışlıkla Dur ve 1'i yanlışlıkla Sağa Dön olarak sınıflandırılmıştır. Dur komutlarından 2 tanesi İlerle olarak yanlış sınıflandırılırken ve İlerle komutlarından 1 tanesi, Dur olarak yanlış sınıflandırılmıştır. Geri Gel komutlarından 1'i yanlışlıkla İlerle olarak sınıflandırılmıştır.

Birleşik veri seti için KA sınıflandırma sonucunun karışıklık matrisi Tablo 7'de verilmiştir.

**Tablo 7. Birleşik veri kümesi için KA algoritmasının karmaşıklık matrisi**

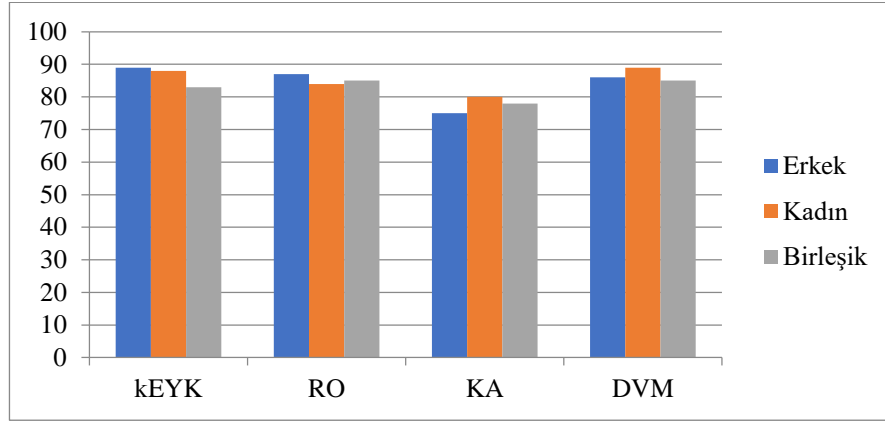
Gerçek \ Tahmin	Dur	İlerle	Sola Dön	Sağa Dön	Geri Gel
Dur	15	1	1		
İlerle	4	18	1		1
Sola Dön	1	1	16		
Sağa Dön			1	20	
Geri Gel			1		19

Tablo 7'ye göre, Sola Dön komutlarından 1'er tanesi diğer sınıflara ait olarak yanlış sınıflandırılmıştır. Dur komutlarından 4 tanesi İlerle olarak yanlış sınıflandırılırken 1 tanesi Sola Dön olarak sınıflandırılmıştır. İlerle komutlarından 1 tanesi Dur olarak, 1 tanesi ise Sola Dön olarak yanlış sınıflandırılmıştır. Geri Gel komutlarından 1'i yanlışlıkla İlerle olarak sınıflandırılmıştır.

Daha fazla analiz yapmak için, veri seti her cinsiyetten 50 yeni ses kaydı ile toplamda 200 ses kaydı olacak şekilde genişletilmiştir. Tüm yöntemler bu genişletilmiş veri seti ile yeniden değerlendirilmiştir. Sonuçlar Şekil 15'te verilmiştir.

Şekil 15'ten görülebileceği gibi, genişletilmiş kadın veri seti için tüm algoritmaların doğrulukları azalmıştır. Genişletilmiş erkek veri seti için KEYK ve RO'nun doğrulukları artarken, KA ve DVM'nin doğrulukları önemli ölçüde azalmıştır.

Birleşik genişletilmiş veri kümesi için kEYK'nun doğruluğu çok fazla farklılık göstermemiştir. Ancak, diğer algoritmaların performansı önemli ölçüde azalmıştır.



**Şekil 15. Genişletilmiş veri seti üzerinde algoritmaların ortalama doğruluk oranları**

Şekil 15'e göre KEYK, erkek veri setinde %89 doğrulukla en yüksek performansa sahipken, kadın veri setinde DVM %89 ile en yüksek doğruluk oranına sahiptir. RO ve DVM, birleşik veri seti için %85 ile aynı en doğru sonuçlara sahiptir.

Algoritmaların performansındaki genel bozulmalar, genişletilmiş veri setinin içeriğinden kaynaklanıyor olabilir. Çünkü yeni ses kayıtları farklı bir zamanda ve farklı çevre koşullarında yapılmıştır. Seslerin perdesi ve tonu, hastalıklar veya çevresel gürültüler nedeniyle etkilenebilir.

Literatürdeki çalışmalar ya ses tanıma performanslarını nicel olarak tartışmamıştır (Bartie ve Mackaness, 2006; Sheldon vd., 2019) ya da kullanıma hazır bir otomatik konuşma tanıma motoru kullanılmıştır (Mirzae vd., 2012; Kanno vd., 2018). Sadece Siyaev ve Jo'nun (2021) yaptığı çalışma performans sonuçları hakkında nicel değerlendirme vermektedir. Bu çalışmada, eğitim veri kümesini elde etmek amacıyla yapay insan sesleri oluşturmak için Google Cloud Text-To-Speech API'yi kullanmışlardır. Eğitim veri seti, 46 farklı yapay hoparlörden elde edilen 45.244 ses dosyasını içermektedir. Önerdikleri model gerçek ses kayıtlarında %94.7 doğruluk oranına sahiptir. Çalışmamızda farklı cinsiyetlerden iki konuşmacının gerçek ses kayıtları ile veri setimiz oluşturulmuştur. 100 kayıtlık ilk birleşik veri setinde DVM ve RO algoritmaları ile %92 en yüksek doğruluk elde edilirken, 200 kayıtlık genişletilmiş birleşik veri setinde doğruluk %85 olarak elde edilmiştir.

## 7. SONUÇ

Deneysel sonuçları genel olarak özetlemek gerekirse, 100 ses kaydı içeren başlangıç veri kümesi için DVM algoritması erkek veri setinde %94 ile en yüksek doğruluğa sahipken, RO algoritması kadın veri setinde %96 ile en yüksek doğruluğu vermiştir. RO ve DVM, birleşik veri seti için %92 ile en yüksek puana sahiptir. Toplam 200 ses kaydı içeren genişletilmiş veri kümesi için ise KEYK, erkek veri setinde %89 doğrulukla en yüksek performansa sahipken, kadın veri setinde DVM %89 ile en yüksek doğruluk oranına sahiptir. RO ve DVM, birleşik veri seti için %85 ile aynı yüksek doğruluğa sahiptir.

Unity'de ses tanıma için kullanılacak farklı ticari kütüphaneler vardır. Bunlardan bazıları; Microsoft'un Windows Speech'i, Google'ın Cloud Speech Recognition ve IBM'in Watson SDK'sı en bilinen ürünlerdir. Ancak, bu ürünleri sesli komut tanıma görevi için kullanmanın maliyet etkin olduğu düşünülmektedir. Ayrıca çevre koşulları veya telaffuz farklılıklarından dolayı zaman zaman ses tanımada da başarısız olabilmektedirler.

Bu çalışmada, AG uygulamalarında sanal nesnelere kontrolü için sesli komut tanıma aracı Python'da hayata geçirilmiş ve Unity'ye entegre edilmiştir. Gelecekteki bir çalışma olarak, ses tanıma modülünün performansı, veri setini farklı kişilere ait daha fazla ses kaydı ile genişleterek iyileştirilecektir.

Performanslarını iyileştirmek için makine öğrenme algoritmalarında farklı ses özellikleri çıkarılabilir ve kullanılabilir. Yeni AG uygulamalarının gereksinimlerine göre ses tanıma yardımcı programına ek komutlar kolayca eklenebilir.

AG uygulamalarında sanal nesnelere sesle etkileşimin, özellikle dokunmatik ekranda hassas konumlandırma gereken durumlarda, kullanım kolaylığı sağlayacağı açıktır. Bunun yanında nörolojik ya da patolojik nedenlerle veya kazalara bağlı olarak ellerini kullanamayan insanlar için de çok önemli kolaylık sunacaktır.

## KAYNAKLAR

- Abdüsselam, M.S., Karal, H. (2012). "Fizik öğretiminde artırılmış gerçeklik ortamlarının öğrenci akademik başarısı üzerine etkisi: 11. Sınıf manyetizma konusu", Eğitim ve Öğretim Araştırmaları Dergisi, 1(4): 170-182.
- ALTINPULLUK, Hakan (2015). "Artırılmış gerçekliği anlamak: kavramlar ve uygulamalar." Açıköğretim Uygulamaları ve Araştırmaları Dergisi 1.4: 123-131.
- Aydın, Ö., (2005). Yapay sinir ağlarını kullanarak bir ses tanıma sistemi geliştirilmesi, Yüksek Lisans Tezi, Trakya Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.
- Bartie, P.J, Mackaness, W.A., (2006). "Development of a Speech-Based Augmented Reality System to Support Exploration of Cityscape", Transactions in GIS, 10(1): 63– 86.
- Baygün, M.K., (2006). Türkçe Komutları Tanıyan Ses Tanıma Sistemi Geliştirilmesi, Yüksek Lisans Tezi, Pamukkale Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı.
- Breiman, L. (2001). "Random Forests", Machine Learning, 45:5–32.
- Cheng, K. H. ve Tsai, C. C., (2013). "Affordances of augmented reality in Science learning: Suggestions for future search." Journal of Science Education and Technology 22.4, 449-462.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.
- Davis, S. and Mermelstein, P. (1980). "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", IEEE Transactions on Acoustics, Speech and Signal Processing, 28, 357-366.
- Dunleavy, M., Dede, C., & Mitchell, R. (2009). Affordances and limitations of immersive participatory augmented reality simulations for teaching and learning. Journal of science Education and Technology, 18(1), 7-22.
- Enyedy, N., Danish, J.A., Delacruz, G. and Kumar, M.. (2012). "Learning physics through play in an augmented reality environment". International journal of computer-supported collaborative learning, 7(3):347-378.
- Esengün, M. (2016). 2 Boyutlu Harita Ve Artırılmış Gerçeklik Tabanlı Mobil Navigasyon Uygulamalarının Kıyaslamalı Değerlendirilmesi (Doctoral dissertation, Fen Bilimleri Enstitüsü).
- İbili, E., Şahin, S. (2013). "Artırılmış Gerçeklik ile İnteraktif 3D Geometri Kitabı Yazılımın Tasarımı ve Geliştirilmesi: ARGE3D", Afyon Kocatepe University Journal of Science and Engineering, 13:1-8.



İçten, T., Bal, G. (2017). "Artırılmış Gerçeklik Üzerine Son Gelişmelerin ve Uygulamaların İncelenmesi", Gazi Üniversitesi Fen Bilimleri Dergisi PART C: TASARIM VE TEKNOLOJİ, 5(2): 111-136.

İnternet: <https://www.layar.com/>, 27.01.2022.

İnternet: <https://www.basebros.com/>, 27.01.2022.

İnternet: <https://www.arimars.com/>, 27.01.2022.

İnternet: <https://digi-mode.com.tr/>, 27.01.2022.

Kaleci, D., Demirel, T., & Akkuş, İ. (2016). Örnek bir artırılmış gerçeklik uygulaması tasarımı. XVIII. Akademik Bilişim Konferansı, Aydın, Türkiye.

Kanno, K.M., Lamounier, E.A., Cardoso, A., Lopes, E.J., and Mendes de Lima, G.F., (2018). "Augmented Reality System for Aiding Mild Alzheimer Patients and Caregivers," 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 593-594.

Kolbe, T. H. (2004). "Augmented videos and panoramas for pedestrian navigation", Proceedings of the 2nd Symposium on Location Based Services & TeleCartography, pp.1-10.

Kurzekar, P. K., Deshmukh, R. R., Waghmare, V. B., & Shrishrimal, P. P. (2014). "A comparative study of feature extraction techniques for speech recognition system", International Journal of Innovative Research in Science, Engineering and Technology, 3(12), 18006-18016.

Liarokapis, F., Petridis, P., Lister, P. F., White, M., (2002). "Multimedia augmented reality interface for e-learning (MARIE), World Transactions on Engineering and Technology Education, 1(2), 173-176.

Livingston, M. A., Ai, Z., Karsch, K., & Gibson, G. O. (2011). User interface design for military AR applications. Virtual Reality, 15(2), 175-184.

McFee, Brian, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. "librosa: Audio and music signal analysis in python." In Proceedings of the 14th python in science conference, pp. 18-25. 2015.

Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1995). Augmented reality: A class of displays on the reality-virtuality continuum. In Telem manipulator and telepresence technologies. Vol. 2351, pp. 282-292ç

Mirzaei, M.R., Ghorshi, S., Mortazavi, M. (2012). "Combining Augmented Reality and Speech Technologies to Help Deaf and Hard of Hearing People", 14th Symposium on Virtual and Augmented Reality, pp.174-181.

Mitchell, T. (1997). "Machine Learning", New York, NY:McGraw-Hill.

Motejlek, J., Alpay, E. (2021). "Taxonomy of Virtual and Augmented Reality Applications in Education", IEEE Transactions on Learning Technologies, 14(3): 415-429.

- Petrović, N. (2020). "Augmented and Virtual Reality Web Applications for Music Stage Performance", 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), pp. 33-36.
- Reitmayr, G., Schmalstieg, D. (2004). "Collaborative Augmented Reality for Outdoor Navigation and Information Browsing", Proceedings of the Second Symposium on Location Based Services and TeleCartography", TU Wien, pp.53-62.
- Azuma, R. (1997). A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments, 6, 355-385.
- Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers.
- Salvador, S., Chan, P., (2004). "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space", KDD Workshop on Mining Temporal and Sequential Data, pp. 70–80, 2004.
- Sheldon, A., Dobbs, T., Fabbri, A., Gardner, N., Haeusler, M.H., Ramos, C. and Zavoleas, Y. (2019). "Putting the AR in (AR)chitecture - Integrating voice recognition and gesture control for Augmented Reality interaction to enhance design practice", Proceedings of the 24th CAADRIA Conference Vol. 1, Victoria University of Wellington, Wellington, New Zealand, 15-18 April 2019, pp. 475-484.
- Sırakaya, M. ve Seferoğlu, S. S., (2016). "Öğrenme Ortamlarında Yeni Bir Araç: Bir Eğitilence Uygulaması Olarak Artırılmış Gerçeklik", Eğitim Teknolojileri okumaları TOJET Sakarya Üniversitesi, 417-438.
- Siyayev, A. Jo, G. (2021). "Neuro-Symbolic Speech Understanding in Aircraft Maintenance Metaverse", IEEE Access, vol.9, pp.154484-154499.
- Somyürek, S. (2014). Öğretim sürecinde z kuşağının dikkatini çekme: artırılmış gerçeklik. Eğitim Teknolojisi Kuram ve Uygulama, 4(1), 63-80.
- Squire, K., & Klopfer, E. (2007). Augmented reality simulations on handheld computers. The journal of the learning sciences, 16(3), 371-413.
- Su, S., Perry, V., Luis, B., Kase, S., Heather, R. Katherine, C., Dasari, V.R. (2020)."Virtual and Augmented Reality Applications to Support Data Analysis and Assessment of Science and Engineering.", Computing in Science & Engineering, 22(3): 27-39.
- Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y. (2008). "Outdoors augmented reality on mobile phone using loxel-based visual feature organization", MIR '08: Proceedings of the 1st ACM international conference on Multimedia information retrieval, pp. 427-434.
- Trujano, F., Khan, Mina, Maes, P. (2018). "ARPIano Efficient Music Learning Using Augmented Reality", International Conference on Innovative Technologies and Learning, pp. 3-17.

Tsai, M.K., Liu, P.H.E., Yau, N.J. (2013). "Using electronic maps and augmented reality-based training materials as escape guide lines for nuclear accidents: An explorative case study in Taiwan", *British Journal of Educational Technology*, 44(1): 18–21.

Wojciechowski, R., Wojciech, C. (2013). "Evaluation of learners' attitude to ward learning in ARIES augmented reality environments", *Computers & Education*, 68, 570-585.

Yoon, C., Louie, R., Ryan, J., Vu, M., Bang, H., Derksen, W., Ruvolo, P. (2019). "Leveraging augmented reality to create apps for people with visual disabilities: A case study in indoor navigation", *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 210-221.

Zainuddin, N. M. M., Zaman, H. B., & Ahmad, A. (2010). Developing augmented reality book for deaf in science: the determining factors. In *IEEE 2010 International Symposium on Information Technology*, pp. 1-4.

Zhang M., (2016). Real-time traffic flow prediction using augmented reality, Yüksek Lisans Tezi, the University of Windsor, School of Computer Science.

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

Adı Soyadı : Tuba EMEKLİ ÇOBAN

### EĞİTİM DURUMU

Lisans Öğrenimi : 2014-2018, KTO Karatay Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü

Yüksek Lisans Öğrenimi : 2021-2023, KTO Karatay Üniversitesi, Mühendislik Fakültesi, Elektrik ve Bilgisayar Mühendisliği Bölümü

Bildiği Yabancı Diller : İngilizce

Sertifikalar : 2021-2022, Karamanoğlu Mehmetbey Üniversitesi, Bilişim Teknolojileri Pedagojik Formasyon Eğitimi

### İŞ DENEYİMİ

Stajlar : 2016, Stajyer Mühendis, Karaman Belediyesi Bilgi İşlem Müdürlüğü

: 2017, Stajyer Mühendis, Karaman Belediyesi Plan Proje Müdürlüğü

Çalıştığı Kurumlar : 2019-2019, Bilgisayar Mühendisi, NovaGlobal Mühendislik Limited Şirketi

: 2022-Halen, Bilgisayar Mühendisi, Tesis Teknoloji A.Ş TÜRKSAT/Karaman

## EK 1. KULLANILAN KODLAR

### 1. Seslerin Kaydedilmesi

```
import numpy as np
import pyaudio
import matplotlib.pyplot as plt
from pydub import AudioSegment
import librosa

fs=8000

def record(duration=2, fs=fs):
    nsamples = duration*fs
    #nsamples = 1024
    p = pyaudio.PyAudio()
    stream = p.open(format=pyaudio.paInt16, channels=1, rate=fs, input=True,
                    frames_per_buffer=nsamples)
    buffer = stream.read(nsamples)
    array = np.frombuffer(buffer, dtype='int16')
    stream.stop_stream()
    stream.close()
    p.terminate()
    return array

print("Kayıt başladı.. 2 sn süresince kayıt edilecektir..")
my_recording = record()

import wavio
wavio.write("myfile.wav", my_recording, fs, sampwidth=2)

data, samplerate = librosa.load('myfile.wav')

plt.figure(figsize=(14,5))
plt.xlabel("Time (msn)")
plt.ylabel("Amplitude (dB)")
plt.plot(data)

audio_format = "wav"
ses = AudioSegment.from_file("myfile.wav", format = audio_format)

from pydub.silence import split_on_silence
audio_chunks = split_on_silence(ses
                                ,min_silence_len = 100
                                ,silence_thresh = ses.dBFS-1
                                ,keep_silence = 50
                                )

file_name = 'sag5'
audio_format = "wav"

combined = AudioSegment.empty()
```

```

for chunk in audio_chunks:
    combined += chunk
combined.export('C:\\Kayitlar\\'+ file_name+'.wav', format = audio_format)

out_file = 'C:\\ Kayitlar\\ '+ file_name+'.wav'

data, samplerate = librosa.load(out_file)

plt.figure(figsize=(14,5))
plt.plot(data)

```

## 2. Seslerin Filtrelenmesi

```

import librosa

out_file = 'C:\\ Kayitlar\\ '+ file_name+'.wav'
data, samplerate = librosa.load(out_file)

from scipy.signal import filtfilt
n = 15
b = [1.0 / n] * n
a = 1
xx = filtfilt(b,a,data)

plt.figure(figsize=(14,5))
librosa.display.waveshow(y=xx,color='b')
plt.xlabel("Time (sn)",fontsize=15)
plt.ylabel("Amplitude (dB)",fontsize=15)
plt.xlim((1,11000))
plt.plot(xx)

```

## 3. kEYK Hariç Diğer Makine Öğrenmesi Algoritmaları için Sonuçların Bulunması

```

import librosa
import numpy as np
import pandas as pd
from scipy.signal import lfilter
from scipy.signal import filtfilt

import os
import re

feature=12
df = pd.DataFrame(columns=['mfcc','lpc','mel', 'mfcc_std','lpc_std','mel_std','komut'])
for root, dirnames, filenames in os.walk("C:\\ Kayitlar\\"):
    for filename in filenames:
        if re.search("\.(wav)$", filename):
            filepath = os.path.join(root, filename)
            #print(filepath)
            x, samplerate = librosa.load(filepath)

```

```

n = 15
b = [1.0 / n] * n
a = 1
xx = lfilter(b,a,x)

mfcc = librosa.feature.mfcc(y=xx, sr=samplerate, hop_length=512, n_mfcc=feature)
features_mfcc = mfcc.T[:,1]
avg_mfcc = np.mean(features_mfcc)
std_mfcc = np.std(features_mfcc)

lpc = librosa.lpc(y=xx, order=4)
features_lpc = lpc[1:]
avg_lpc = np.mean(features_lpc)
std_lpc = np.std(features_lpc)

melspec = librosa.feature.melspectrogram(y=xx, sr=samplerate, hop_length=512)
features_mel = melspec
avg_mel = np.mean(features_mel)
std_mel = np.std(features_mel)

a = [avg_mfcc, avg_lpc, avg_mel, std_mfcc, std_lpc, std_mel]

if (filepath.find("dur")!=-1):
    a=np.append(a,[1],axis=0)
if (filepath.find("leri")!=-1):
    a=np.append(a,[2],axis=0)
if (filepath.find("sol")!=-1):
    a=np.append(a,[3],axis=0)
if (filepath.find("sag")!=-1):
    a=np.append(a,[4],axis=0)
if (filepath.find("geri")!=-1):
    a=np.append(a,[5],axis=0)
# print(features_x)
df=df.append(pd.DataFrame(a.reshape(1,-1), columns=list(df)),ignore_index=True)

df.to_csv("allfeatures.csv",index=False)
datalar = pd.read_csv('allfeatures.csv')

float_formatter = "{:.2f}".format
np.set_printoptions(formatter={'float_kind':float_formatter})
print(datalar)
komutlar = datalar["komut"]
print(komutlar)

datalar = datalar.drop("komut",axis=1)

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(datalar)

import pickle
pickle.dump(scaler,open('scaler.sav', 'wb'))

```

```

from sklearn.model_selection import train_test_split
train_set, test_set, train_labels, test_labels = train_test_split(scaled_data, komutlar,
test_size=0.2, random_state=42)

kucuk_train = train_set
kucuk_labels = train_labels

train_set = np.concatenate((train_set, test_set))
train_labels = np.concatenate((train_labels, test_labels))

from sklearn.ensemble import RandomForestClassifier
forest_clf = RandomForestClassifier(random_state=42)

from sklearn.tree import DecisionTreeClassifier
dt_clf = DecisionTreeClassifier(random_state=0)

from sklearn import svm
svm_clf = svm.SVC()

from sklearn.model_selection import cross_val_score
forest_scores = cross_val_score(forest_clf, train_set, train_labels, cv=3, scoring='accuracy')
dt_scores = cross_val_score(dt_clf, train_set, train_labels, cv=3, scoring='accuracy')
svm_scores = cross_val_score(svm_clf, train_set, train_labels, cv=3, scoring='accuracy')

print("Forest Cross validation scores")
print(np.mean(forest_scores))
print(forest_scores)

print("DT Cross validation scores")
print(np.mean(dt_scores))
print(dt_scores)

print("SVM Cross validation scores")
print(np.mean(svm_scores))
print(svm_scores)

forest_clf.fit(kucuk_train, kucuk_labels)
dt_clf.fit(kucuk_train, kucuk_labels)
svm_clf.fit(kucuk_train, kucuk_labels)

file_svm = 'svm.sav'
file_forest = 'randomforest.sav'
file_dt = "dt.sav"

pickle.dump(svm_clf, open(file_svm, 'wb'))
pickle.dump(forest_clf, open(file_forest, 'wb'))
pickle.dump(dt_clf, open(file_dt, 'wb'))

dt_clf = pickle.load(open(file_dt, 'rb'))
forest_clf = pickle.load(open(file_forest, 'rb'))
svm_clf = pickle.load(open(file_svm, 'rb'))

```



```

result_svm = svm_clf.score(test_set, test_labels)
result_forest = forest_clf.score(test_set, test_labels)
result_dt = dt_clf.score(test_set, test_labels)

print("Forest Test set results")
print(result_forest)

print("DT Test set results")
print(result_dt)

print("SVM Test set results")
print(result_svm)

print("#### Cross Validation Predict Results ####")

from sklearn.model_selection import cross_val_predict
from sklearn.metrics import confusion_matrix

svm_tahmin = cross_val_predict(svm_clf,train_set,train_labels,cv=3)
svm_cm = confusion_matrix(train_labels,svm_tahmin)
print(svm_cm)

forest_tahmin = cross_val_predict(forest_clf,train_set,train_labels,cv=3)
forest_cm = confusion_matrix(train_labels,forest_tahmin)
print(forest_cm)

dt_tahmin = cross_val_predict(dt_clf,train_set,train_labels,cv=3)
dt_cm = confusion_matrix(train_labels,dt_tahmin)
print(dt_cm)

from sklearn.metrics import precision_score, recall_score,f1_score

print("RF Metrics")
print("Precision", precision_score(train_labels,forest_tahmin,average='micro'))
print("Recall", recall_score(train_labels,forest_tahmin,average='micro'))
print("f1_score", f1_score(train_labels,forest_tahmin,average='micro'))

print("DT Metrics")
print("Precision", precision_score(train_labels,dt_tahmin,average='micro'))
print("Recall", recall_score(train_labels,dt_tahmin,average='micro'))
print("f1_score", f1_score(train_labels,dt_tahmin,average='micro'))

print("SVM Metrics")
print("Precision", precision_score(train_labels,svm_tahmin,average='micro'))
print("Recall", recall_score(train_labels,svm_tahmin,average='micro'))
print("f1_score", f1_score(train_labels,svm_tahmin,average='micro'))

```

#### 4. KEYK Algoritması için Sonuçların Bulunması

```
import librosa
import numpy as np
import pandas as pd

from scipy.signal import lfilter
from scipy.signal import filtfilt

feature=24

df = pd.DataFrame(index=range(29))

import os
import re

uzun = 0
for root, dirnames, filenames in os.walk("C:\\Kayitlar\\ "):
    for filename in filenames:
        if re.search("\.(wav)$", filename):
            filepath = os.path.join(root, filename)
            print(filepath)
            x, samplerate = librosa.load(filepath)
            n = 15
            b = [1.0 / n] * n
            a = 1
            xx = filtfilt(b,a,x)
            mfcc = librosa.feature.mfcc(y=xx, sr=samplerate, hop_length=512,
n_mfcc=feature)
            features_x = mfcc.T[:,1]

            if (filepath.find("dur")!=-1):
                features_x=np.append(features_x,[1],axis=0)
            if (filepath.find("leri")!=-1):
                features_x=np.append(features_x,[2],axis=0)
            if (filepath.find("sol")!=-1):
                features_x=np.append(features_x,[3],axis=0)
            if (filepath.find("sag")!=-1):
                features_x=np.append(features_x,[4],axis=0)
            if (filepath.find("geri")!=-1):
                features_x=np.append(features_x,[5],axis=0)

            df[filename]=pd.Series(features_x)
            if (len(features_x))>uzun:
                uzun=len(features_x)

print(uzun)

df.to_csv("mfcc.csv")

print("***** Dataframe *****")
```

```

from fastdtw import fastdtw
from scipy.spatial.distance import euclidean

df = pd.read_csv('mfcc.csv')

df = df.drop(df.columns[[0]],axis = 1)

import operator

float_formatter = "{:.2f}".format
np.set_printoptions(formatter={'float_kind':float_formatter})

sonuclar = []
basarisiz = 0
sayi=0
for i in range(50):
    sonuc = []

    b = []
    A=df.iloc[:,i].array
    A = A[~np.isnan(A)]
    A_sinif=A[-1]
    A = A[:-1]
    for j in range(50):
        a = []
        if (i!=j):
            B=df.iloc[:,j].array
            B = B[~np.isnan(B)]
            B_sinif=B[-1]
            B = B[:-1]
            dist, path = fastdtw(A, B, dist=euclidean)
            a = [dist,A_sinif,B_sinif]
            b=np.concatenate((b,a))
    b=b.reshape(-1,3)
    c=b[b[:, 0].argsort()]

    if (sayi!=A_sinif) and (sayi!=0):
        sonuc = [sayi,basarisiz]
        sonuclar=np.concatenate((sonuclar,sonuc))
        basarisiz=0

toplam=0
sayi=c[0,1]
for ii in np.arange(6):
    satir=c[ii]
    if (sayi!=satir[2]):
        print("sayi:",sayi,"asil_sayi:",satir[2])
    if (sayi==satir[2]):
        toplam=toplam + 1
print("Sayı:",sayi)
print("-----")
print("Toplam doğru: ",toplam)

```

```
print("Doğru oran: ", toplam/6)
if ((toplam/6<0.5) and (sayi==A_sinif)):
    print("sayi",sayi,"A_sinif",A_sinif)
    basarisiz=basarisiz+1
    print("basarisiz",basarisiz)

sonuc = [sayi,basarisiz]
sonuclar=np.concatenate((sonuclar,sonuc))
print(sonuclar)
```