

**T.C.
FIRAT ÜNİVERSİTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**



**FLUTTER MOBİL UYGULAMA GELİŞTİRME ORTAMINDA
YAPAY ZEKÂ TABANLI PARA TANIMA SİSTEMİ**

Baraa KALAAJI – Yousif AL-EZZI

HAZİRAN - 2022

T.C.
FIRAT ÜNİVERSİTESİ
YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ

Başlığı:	Flutter Mobil Uygulama Geliştirme Ortamında Yapay Zekâ Tabanlı Para Tanıma Sistemi
Yazarı:	Baraa KALAAJI, Yousif AL-EZZI
Proje Danışmanı:	Doç. Dr. Fatih ÖZKAYNAK
Teslim Tarihi:	12.Haziran.2022

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER.....	i
ÖZET	ii
ŞEKİLLER LİSTESİ.....	iii
1. GİRİŞ	1
2. GEREKSİNİM TASARIMI VE ANALİZİ	3
3. SİSTEM TASARIMI	4
4. SİSTEM GERÇEKLEŞTİRİMİ	1
5. MATERYAL VE METOT.....	1
6. BULGULAR VE TARTIŞMA	1
7. SONUÇLAR	1
KAYNAKLAR	2

ÖZET

Flutter Mobil Uygulama Geliştirme Ortamında Yapay Zekâ Tabanlı Para Tanıma Sistemi

Baraa KALAAJI, , Yousif AL-EZZI

FIRAT ÜNİVERSİTESİ
Yazılım Mühendisliği Bölümü

Yapay zekâ, teknolojik çağında yaşayan insanların hayatlarını şekillendiren ve şirketlerin yapısından ayrılmaz bir parça haline gelen bir kavramdır. Bunun için yüksek verimlilik ve performans değerlerine sahip olan TensorFlow Lite Kütüphanesi, Flutter gibi mobil uygulamaları geliştirme ortamında gerçekleştirmek çok yüksek bir önem taşımaktadır.

Proje içerisinde Kaggle platformundan alınan veri seti kullanarak eğitilen yapay zekâ modeli, yüksek performans ve doğruluk değerleri ile Türk lirası (TL) para birimi tanıyabilmesi var sayılmıştır.

Flutter ortamında kullanılacak para tanıma yapay zekâ modeline performans ve uyumluluk testleri geçirerek model, düşük donanıma sahip olan telefonlarda (Low End Devices) bile performanslı bir şekilde çalışabileceği gözlenmiştir.

Para tanınması, minimalist kullanıcı ara yüzü (User Interface - UI) elementlerine sahip kamera uygulaması Zoom, Işık ve Auto Flash özellikleri ile farklı kullanım koşulları altında gerçekleştirebilmesi var sayılmıştır.

Proje, Flutter ortamında çalışan farklı para tanıma uygulamaları ile mukayese ederek yüksek performans ile çalışabileceği tespit edilmiştir.

Anahtar Kelimeler: Yapay Zekâ, TensorFlow Lite, Para Tanıma Sistemi, Flutter Ortamında Yapay Zekâ

ŞEKİLLER LİSTESİ

Sayfa

Şekil 2.1.	Bir kaldırım üzerinde konulan uzun çizgilere sahip sarı şerit... Error! Bookmark not defined.	
Şekil 2.2.	Normal Beyaz Baston Error! Bookmark not defined.	
Şekil 2.3.	WeWalk – Elektronik beyaz baston Error! Bookmark not defined.	
Şekil 3.1.	Use-Case Diyagramı6	
Şekil 3.2.	Sistem mimarisi diyagramı.....7	
Şekil 3.3.	Kullanıcı Arabirimleri.....9	
Şekil 5.1.	YOLOv3 diğer nesne tespiti algoritmaları ile kıyaslama1	
Şekil 5.2.	YOLOv2 Tiny ağ yapısı.....2	
Şekil 5.3.	YOLOv2 Tiny modeli ile tespit edilen bir nesnenin Bounding Boxu5	
Şekil 5.4.	Kullanıcı Arayüzü.....6	

1. Giriş

Birçok toplantı için toplantının refah düzeyi o toplantıdaki bulunan engellilere bakım ve onlara yardım edebilecek girişimlerin sayısı ile doğru orantılıdır. Özellikle son elli yıllık süreçte hayatımızın ayrılmaz bir parçası haline gelen yeni yazılımsal ve donanımsal teknolojileri kullanarak engellilerin yardımına yönelik projelerin ve araçların geliştirmelerinin fırsatı gün geçtikçe artmaktadır.

Dünya çapında bir milyar insan - dünya nüfusunun %15'i - bir tür engellilik yaşıyor. Engellilik, aramızdaki neredeyse her altı kişiden birini etkilemesine rağmen, engellilerin çoğu, uygun bir fiziksel çevre, devlet hizmetleri, yardımcı cihazlar ve teknolojilerin olmaması nedeniyle hala sosyal ve ekonomik fırsatlara tam olarak katılmaktan dışlanmış durumdadır [1]. Bu sayının %21'i görme engellilerden oluşmaktadır. Bir tür görme engelliğine sahip olan 284 milyon kişiden 39'ı tamamen kör insanlardır [2].

Türkiye'de ise Ulusal Engelli Veri Tabanına göre engelli birey sayısı 1.559.222. (Ancak resmi olmayan rakamlara göre %13 düzeyinde, 9 milyon olduğu söyleniyor) Bunların %27'si 0-21 yaş, %36'sı 22-49 yaş, %37'siyse 50-64 yaş arasındadır. Bu kişiler içerisinde 215 bin 76'sını görme engelliler oluşturmaktadır [3].

Görme engelli bireylerin yaşantılarında, eğitimlerinde veya çalışma alanlarında, onlar için tasarlanmış araçlar bulunmaktadır. Bu araçlar mobil ve akıllı saat gibi taşınabilir cihazların gelişmesiyle, yapay zekâ alanında yeni performanslı tekniklerin keşfedilmesi ile gelişmektedir. [4].

Son zamanlarda çok önemli ve popüler bir konu haline gelen yapay zekâ, hayatımızda düzenli bir şekilde kullandığımız çeşitli elektronik aletlerden temel bir parçası oluşturmaktadır. Uzman sistemler ile başlayan yapay zekâ uygulamaları, bir problemi o problemin uzmanlarının çözdüğü gibi çözeabilen bilgisayar programları geliştiren teknolojidir. Uzmanlar problemleri çözerken bilgilerini ve deneyimlerini kullanmaktadırlar. Bu bilgi ve deneyimlerin bilgisayar tarafından anlaşılabilir olması ve bilgisayarda saklanması gerekmektedir. Bilgi tabanında saklanan bu bilgileri kullanarak insan karar verme sürecine benzer bir süreç ile problemlere çözüm üretmektedir. Kısacası uzman sistemler bilgi tabanlı sistemler olup, problemleri daha geniş bir çerçevede inceleyip, çözümünde insan zekâsını taklit etmeyi hedefleyen yapay zekânın bir uygulamasıdır.

Uzmanın (insanın) düşünme ve karar alma süreci bilgisayar tarafından bire bir taklit etmesi yapay zekâ ile çözülebilen problemlerin alanı çok dar hale getirmektedir. Bunun için bu alanı büyütmek için insanın problem çözme mantığı bilgisayar tarafından daha efektif bir şekilde kullanabilmek için yeni bir yaklaşıma ihtiyaç duyulmaktadır.

Problemin çözümü için belirsizliğin oldukça önemli olduğunun ortaya çıkmasından ötürü, geleneksel mantıktan belirsizliği baz alan bir düşünceye geçişini dönüşümünü başlatmıştır. Bu

düşünce ilk olarak klasik mantığa aykırı olduğundan ötürü olumlu yaklaşılmaya da bu dönem içerisinde belirsizliğin bazı alan yöntemler üzerinde çalışıldığını görülmüştür. Bulanık Mantık, ilk kez Azerbaycanlı Prof. Lutfi A. Zadeh tarafından ortaya atılmıştır. Zadeh çalışmasında belirsizliğin kesinlik anlamı içermeyen sınırlar ile ifade ederek bulanık küme mantığını ortaya koymuştur. İnsanlardan karşısına günlük hayatta çıkan problemlerin birçoğu belirsizlik içerdiğinden ötürü, bulanık mantık bu tür problemleri üyelik dereceleri ile anlamlaştırarak hangi duruma ne kadar ait olduğunu belirlemektedir. Günümüzde karmaşık problemlerin çözümünde, bulanık mantık uygulanabilir kullanışlı bir tercihtir ancak Uzman Sistemler gibi bulanık mantık sistemlerinin bulanık denetimde kullanılan kurallar uzman bilgi ve tecrübesine bağlı olduğundan dolayı problemlerin çözümleri yapay zekâdan beklendiği gibi olmayıp tüm uygulamalar insanın zekâsına bağlı kalmıştır.

Bu problemi çözmek için Yapay Sinir Ağları (YSA) geliştirilmiştir. Yapay sinir ağları, insanın beyninin özelliklerinden olan öğrenme yolu ile yeni bilgiler üretebilme, yeni bilgiler oluşturabilme ve keşfedebilme gibi yetenekleri ilgili alandaki veri örneklerinden otomatik olarak gerçekleştirmek amacı ile geliştirilen bilgisayar yazılımlarıdır.

Beyni bir işlemci ağına benzetilirse, yapılması gereken problemin işlemci ağına dağıtılması ve yerel parametrelerin belirlenmesidir. Yapay sinir ağlarının öğrenme yeteneği bu işlemden kazanılmaktadır.

Yapay sinir ağlarından başlayarak ve yapay zekânın algoritmalarını ve kullanılan metotları geliştirerek yapay zekâ yazılımları çok daha efektif hale gelmiştir.

Son zamanlarda, özellikle yapay zekâ alanında geliştirilen algoritmaların çözüm üretme etkililiği çok yüksek olduğunda görüntü işleme gibi bazı yapay zekâ uygulamaları çok popüler bir hale gelmiştir. Görüntülerden nesnenin tespitini yapmada kullanılan görüntü işleme algoritmaları ile yapay zekanın alt dallarından biri olan derin öğrenme çeşitli donanımlar ile entegre ederek otonom insansız araçları, medikal görüntülerden çeşitli hastalıkların tespiti ve engelli bireyler için asistan teknolojiler gibi bir çok alanda uygulama geliştirilmektedir.

Flutter mobil uygulama geliştirme ortamında Yapay Zekâ tabanlı para tanıma sistemi projesi ise bu görüntü işleme ve tespiti algoritmaları kullanarak kamerada gelen görüntülerin analizini yaparak kullanıcının tuttuğu para birimi tanımasına yardımcı olacaktır. Bunun için görüldüğü üzere projenin kapsamı az görme ve kör insanlardan oluşmaktadır.

2. GEREKSİNİM TASARIMI VE ANALİZİ

Yazılım mühendisliği projelerinin başlangıç noktası olarak bilinen Gereksinim Tasarımı ve Analizi aşaması yazılım projenin en önemli aşamasıdır. Bu aşamada yapılan araştırma ve analiz çalışmaları projenin potansiyel başarısı ve gerçek hayatta etkinliği hakkında ön fikir oluşmaktadır.

2.1. Mevcut Sistem İncelemesi

Mevcut sistem incelemesinde Türk lirası tanıyabilen uygulamaları var olduğunu görülmektedir. Bu uygulamaların analizini ve incelemesini yaparak bu çalışma kapsamında gerçekleştirilecek proje ile mukayese edilmesini sağlanacaktır.

2.1.1. Python İle Para Tanıma Sistemi

Yapay Zekâ uygulamaları geliştirmesinde sıkça kullanılan Python dili, yapay zekâ uygulamaları açısından zengin bir dildir. Python dili ile yazılan ve OpenCV kütüphanesi kullanan Türk lirası para tanıma sistemi, aynı misyon ile geliştirilen yapay zekâ modellerinden daha iyi performans göstermesine rağmen, Python dili ile yazılmış olması kullanıcıların modele erişmesine olumsuz bir şekilde etkilemektedir. Python dili ile yazılan yapay zekâ modellerin iki kullanma ve uygulama yolu vardır:

- **Bilgisayar üzerinde çalışmak:** Az görme engellilerin bilgisayar kullanımı sınırlı ve zor olmasından dolayı bilgisayar üzerinde çalıştırılan yapay zeka modellerine erişimlerini de zor olmaktadır.
- **Sunucuya yükleyerek Web sitesinde çalıştırmak:** Bu şekilde çalışan yapay zekâ modellerin kullanabilmesi için sürekli internet erişime ihtiyaç bulunmaktadır. Aynı zamanda görme engellilerin para tanıma sistemi gibi sıkça kullanacağı bir sistemin web sayfası her bir kullanım durumunda tekrar açmak, bu uygulama yolu da proje için uygun olmadığı göstermektedir.

2.1.2. Erişilebilirlik Hayat Uygulaması

Erişilebilir Yaşam Mobil Uygulaması engellilerin günlük hayata katılımlarına katkı sağlayan, engellilerin günlük hayatta yaşadığı sorunlara somut çözümler getiren çeşitli servisleri bir arada sunan tüm operatörler için ücretsiz bir iOS ve Android mobil uygulamasıdır. Bu uygulama Para Tanıma, metin seslendirme, alt yazı ekleme gibi engellilerin kullanabileceği tüm fonksiyonlar bir yer içinde toplamıştır. Bu proje, mevcut sistemler arasında en fonksiyonlu uygulamadır. Ancak

uygulamayı kullanabilmek için Türk Telekom müşterisi olmak zorunlu bir şart olduğundan bu proje, görme engelliklerine sahip olan tüm vatandaşlar tarafından kullanmasında ciddi bir mâni haline gelmektedir.

2.1.3. Mevcut Sistemin Değerlendirilmesi

Mevcut sistem incelemesin bölümünde görüldüğü gibi mevcut halde bulunan bazı sistemler sınırlıdır ve negatif yönleri de sahiptir. Bunun için bu çalışma kapsamında önerilen sistem bu negatif yönleri çözmeye hedeflemektedir.

2.2. Önerilen Sistem Analizi

Mevcut sistem ve araçları incelendiği zaman, sistemin eksikleri gözükmemektedir. Bunun için bu çalışma kapsamında önerilen sistemin analizini yaparak mevcut sistemdeki bulunan deavantajları nasıl ortadan kaldıracakları gösterilecektir.

2.2.1. Yapay Zekâ Tabanlı Para Tanıma Sistemi

Proje bünyesinde gerçekleştirmek hedeflediğimiz para tanıma sistemi temel olarak yapay zekâ modeli kullanarak çalışmaktadır. Kaggle platformundan alınan 5000 görüntüden fazla sahip Türk lirası veri seti ile eğitilen yapay zekâ modeli, TensorFlow Lite algoritması ile mobil gibi taşınabilir cihazlarda yüksek performans ve doğruluk oranı sağlamaktadır. Aynı zamanda eğitilen yapay zekâ modeli, Flutter framework’u içinde uygulayarak projemizi düşük donanımlara sahip olan telefonlarda bile çalışabilecektir.

Projemizi çalıştırmak için gerekli olan paranın görüntüsü alabilmek amacıyla özel kamera uygulaması geliştirilmiştir. Kamera uygulamasına Zoom, Auto Flash ve Işık ayarları ekleyerek paranın görüntüsü farklı hava ve ışık koşulları altında alınabilmesi ve yapay zekâ modeli tarafında tanınabilmesi için yol sağlanmaktadır.

Projenin özelliklerine ziyade proje Play Store ve App Store uygulamalarında tüm özellikleri ile bedava olması

3. SİSTEM TASARIMI

3.1. Genel Tasarım Bilgileri

3.1.1. İşlevsel Belirtiler

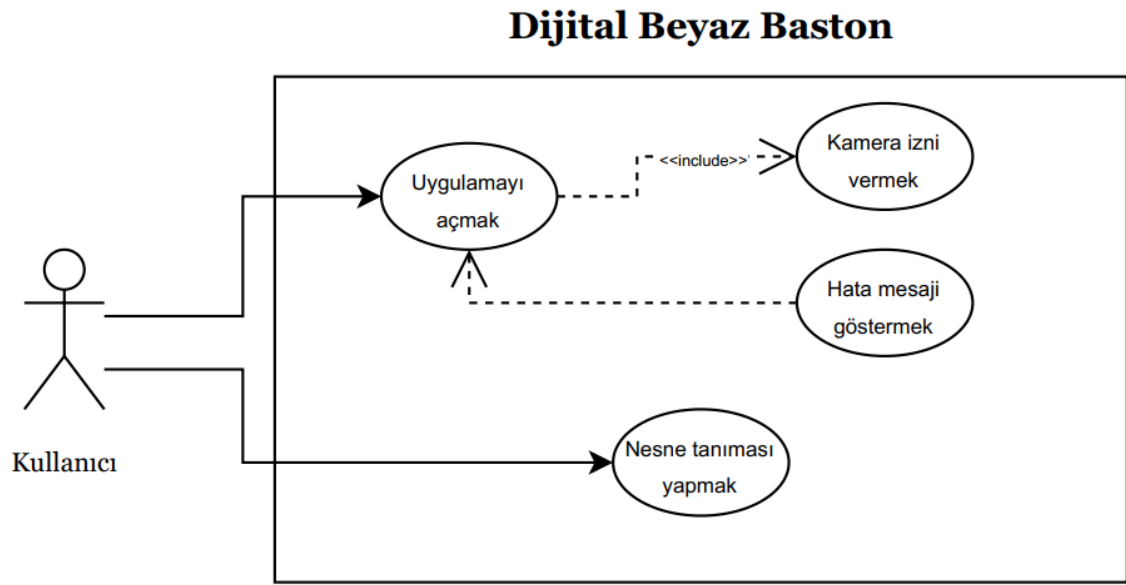
İşlevsel belirtilerin tasarımı, sistemin ve yazılımın temel görevini tanımlayarak belirlenmektedir. Sistem görme engellilerin dolaşımı rahat bir şekilde sağlayan nesne tespiti özelliği birden fazla zayıf noktası olan normal ve elektronik beyaz bastondan türeteren daha etkin bir nesne tespiti yolu sunmaktadır. Yazılım ise arka kameradan alınan görüntü üzerinde önceden TensorFlow yapay zeka kütüphanesiyle eğitilen modeli kullanarak nesne tespiti yapılmaktadır. Tanınması yapılan nesnelerin isimlerini Google metinden konuşmaya motoru (TTS Engine) kullanarak sesli olarak söylemektedir.

3.1.2. Varsayımlar Ve Kısaltmalar

Sistemde varsayılan değerler bulunmamakta bunun yanı sıra başlıca kullanılacak cihazın donanımı ve yazılımı hakkında kısaltmalar vardır. Kısaltma olarak belirlenen donanım ve yazılım sınırlarına sahip olmayan mobil cihazları dijital beyaz bastonun uygulaması mümkün olmayacaktır.

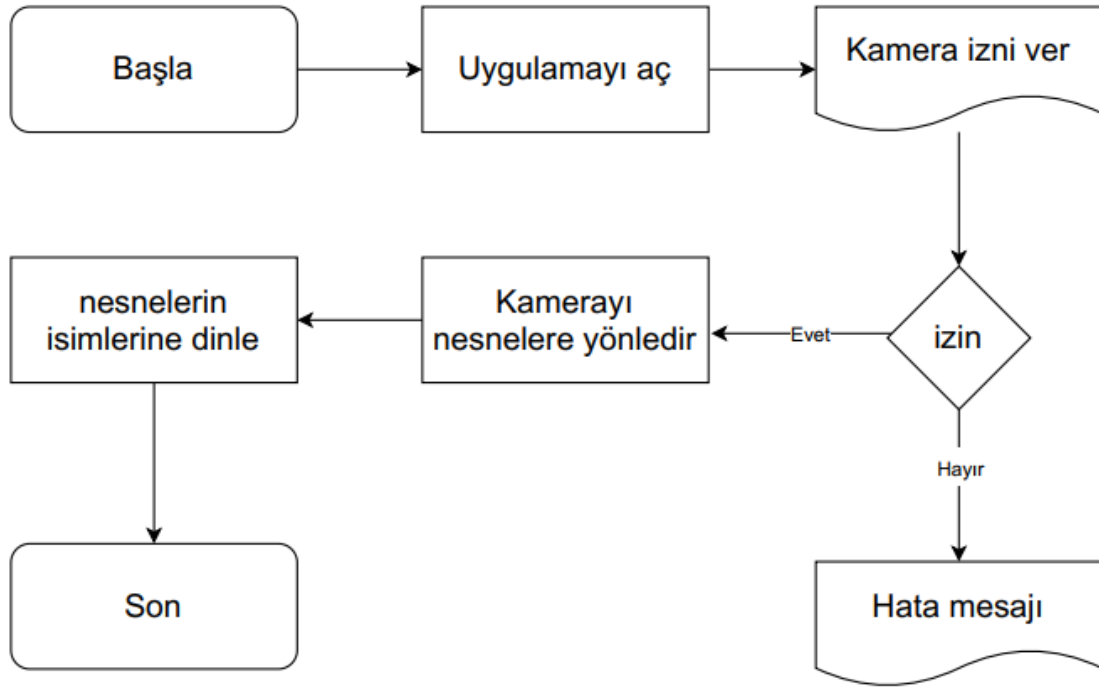
- Donanım Kısaltmalar (minimum):
 - 150 MB Depolama yeri
 - 3 GB Ram
- Yazılım Kısaltmalar (minimum):
 - Android OS

3.1.3. Sistem Mimarisi



Şekil Error! No text of specified style in document..1.

Use-Case Diyagramı



Şekil Error! No text of specified style in document..2.

Sistem mimarisi diyagramı

3.1.4. Testler

Genel hatlarıyla test iki aşamada gerçekleştirilecektir. Bilinen adıyla pilot bölge uygulaması yapılacaktır.

Alfa Aşaması: Sistemin geliştirildiği yerde kullanıcıların gelerek katkıda bulunması sistemi test etmesi ile yapılacaktır.

Beta Aşaması: Kullanıcı, geliştirilen sistemi kendi yerleşkesinde, bir gözetmen eşliğinde yapılacaktır.

3.1.5. Performans

Uygulama içinde gerçek zamanlı (Real Time) olarak çalışacak yapay zeka modelin performansı, genel uygulamanın performansında çok kritik bir role sahiptir. Modelin nesne tespiti hızı, saniyede kare sayısı (Frame Per Second - FPS), v.b. parametrelerin performans üzerinde etkisi büyüktür.

Sistemin performansını etkileyen faktörlerin test verileri değerlendirilecektir. Aynı zamanda sistemin tasarıma uygunluk performansı kontrol ederek tasarımı yapılan sistemin stabilitesi ve işleyiş performansı değerlendirilecektir.

3.2. Süreç Tasarımı

Süreç tasarımı yapar iken uygulamanın içerisinde kullanılacak farklı modüller detaylı bir şekilde yazılmaktadır. Ancak bu çalışma kapsamında gerçekleştirilecek proje içerisinde kullanıcı modülü hariç yönetici ve veri tabanı modülleri gibi farklı bir modül bulunmadığından dolayı sadece kullanıcı modülü incelenmektedir.

3.2.1. İşlev

Kullanıcının sisteme müdahale edebileceği ekrana erişmesi ve uygulamayı kullanabilmesi için aşması gereken bir modüldür.

3.2.2. Kullanıcı Arabirimi



Şekil Error! No text of specified style in document...

3.2.3. Modül İç Tasarımı

Proje içinde kullanıcı modülü hariç başka modül kullanılmadığından dolayı modülün iç tasarımı Şekil 3.2.'teki sistemin mimarisine denk gelmektedir.

4. SİSTEM GERÇEKLEŞTİRİMİ

Gerçekleştirim çalışması, tasarım sonucu üretilen süreç bilgisayar ortamında çalışan yazılım biçimine dönüştürülmesi çalışmalarını içermektedir.

4.1. Yazılım Geliştirme Ortamları

4.1.1. Programlama Dilleri Ve Frameworklar

Dart: Dart, ilk kez Google tarafından geliştirilen açık kaynaklı ve genel-amaçlı bir programlama dilidir. Dart dili kullanılarak web, sunucu, mobil uygulamalar ve IoT cihazları geliştirilebilir. Dart dili sınıf-temelli, tekil-kalıtımlı C-tarzında bir kod dizilimine sahiptir ve JavaScript diline veya çalıştığı sistemdeki özgün (native) dile çevrilebilir. Google, Dart'ın tüm kısımlarının Android ve iOS üzerinde native mobil uygulama geliştirme yapılabilmesi için Flutter üzerinde uğraşmaktadır. [5]

Flutter: Flutter, Google tarafından oluşturulan açık kaynaklı bir UI yazılım geliştirme kitidir. Android, iOS, Windows, Mac, Linux ve web için uygulamalar geliştirmek için kullanılmaktadır. Flutter, tam olarak işleyen bir uygulama sunmak için çeşitli widget'ler kullanmakta ve bu widget'ler Flutter'ın çerçeve mimarisi oluşturmaktadır. [6]

Python: Python, nesne yönelimli, yorumlamalı, birimsel (modüler) ve etkileşimli yüksek seviyeli bir programlama dilidir. Modüler yapısı, sınıf dizgesini (sistem) ve her türlü veri alanı girişini desteklemektedir. Python ile sistem programlama, kullanıcı arabirimi programlama, ağ programlama, web programlama, uygulama ve veri tabanı yazılımı programlama gibi birçok alanda yazılım geliştirilebilmektedir. [7]

TensorFlow: TensorFlow, makine öğrenimi için ücretsiz ve açık kaynaklı bir yazılım kütüphanesidir . Bir dizi görevde kullanılabilir, ancak derin sinir ağlarının eğitimi ve çıkarımına özel olarak odaklanmaktadır. Tensorflow, veri akışına ve türevlenebilir programlamaya dayalı sembolik bir matematik kitaplığıdır. Google'da hem araştırma hem de üretim için kullanılmaktadır.[8]

Mobil uygulamanın gerçekleştirmesinde Dart programlama dili kullanan Flutter frameworku kullanılmaktadır. Flutter, çok iyi dökümantasyona sahip olmasından, hibrit mobil uygulama geliştirme frameworklarına göre daha hızlı olmasından, native programlama dilleri ile geliştirilen

uygulamaların performansına yakın bir performansa sahip olmasından ve TensorFlow ile eğitilen modellerin kolay bir şekilde kullanımı sağlamasından bu proje için tercih edilmektedir.

4.1.2. Yazılım Geliştirme Ortamı

Flutter framework kullanabilmek için Flutter'ı destekleyen bir geliştirme ortamı hazırlanması ve kullanması gerekmektedir. Bu proje kapsamında geliştirme ortamı olarak Microsoft tarafından geliştirilen hafif ve güçlü VS Code kaynak kodu düzenleyicisi (editor) kullanılmaktadır.

4.2. Kod Yapılandırması

4.2.1. Açıklama Satırları

Açıklama satırları kodun yazılması, test ve bakım edilmesi kolaylaştırdığından dolayı bu projede düzgün ve düzenli bir şekilde yazılmasına özen gösterilmiştir. Açıklama satırları karmaşık ve programın akışına büyük etkisi her satırın ve metodun başında yazılmaktadır.

4.2.2. Anlamlı İsimlendirme

Sistem kodlamasının genel yapısında kullanılan değişkenlerin ve metodların isimlerini, kodun içindeki işlevini açık bir şekilde gösterecek şekilde ve büyük küçük harflerin duyarlılığına dikkat ederek yazılmaktadır.

4.2.3. Kod Biçimlemesi

Kod biçimi tasarlanır iken kodun anlaşılabilirliği ve kullanılabilirliğini artmaya hedeflenmektedir. Bunun için kod biçimlemesinde 4 temel kısım bulunmamaktadır:

- **Base:** Kodun temel kaynaklarını içeren kısımdır. Bu kısım içinde kodun içinde kullanılan renk, yazı tipi, yazı boyutu, yazı kalınlığı, v.b. bulunmaktadır. Aynı zamanda program çalışırken uygulamanın farklı sayfalarına yönlendirme sistemi de bulunmaktadır.
- **Models:** Yapay zeka modeli kullanıldığı kısımdır. Bu kısım içinde yapay zeka modelin ürettiği çıktıları elde edilmekte ve kullanıma hazır olmaktadır.
- **Pages:** Uygulama içinde kullanılan farklı sayfaların tasarımın yapıldığı kısımdır. Bu kısım içinde her bir sayfanın widget yapısı bulunmaktadır.
- **Services:** TensorFlow ve metinden konuşmaya (TTS) kütüphaneleri gibi kullanıldığı kısımdır.

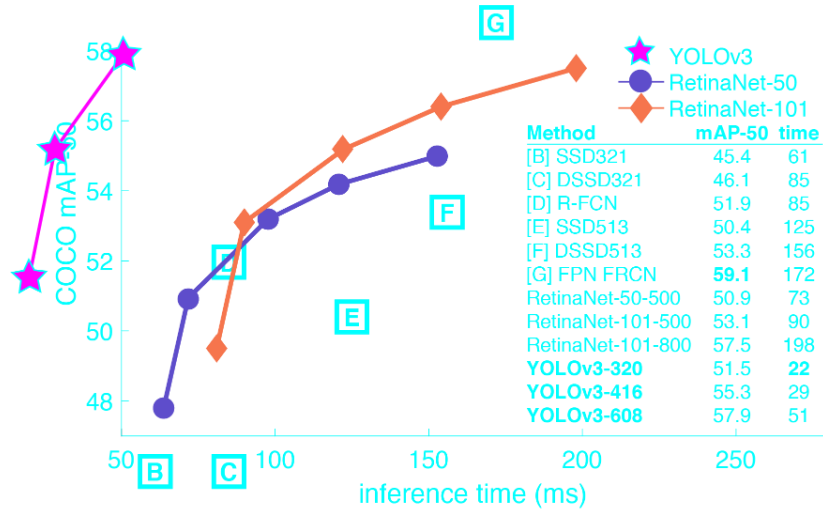
4.3. Olağandışı Durum Tanımları

Olağan dışı durum, bir programın çalışmasının, geçersiz ya da yanlış veri oluşumu ya da başka nedenlerle istenmeyen bir biçimde sonlanmasına neden olan durum olarak tanımlanmaktadır. Olağandışı gelişen durumlarda try-catch blokları devreye girmekte ve program kırılmadan çalışmasına devam edebilecek şekilde kod yazmaya özen gösterilmektedir.

5. MATERYAL VE METOT

5.1. YOLO Algoritması

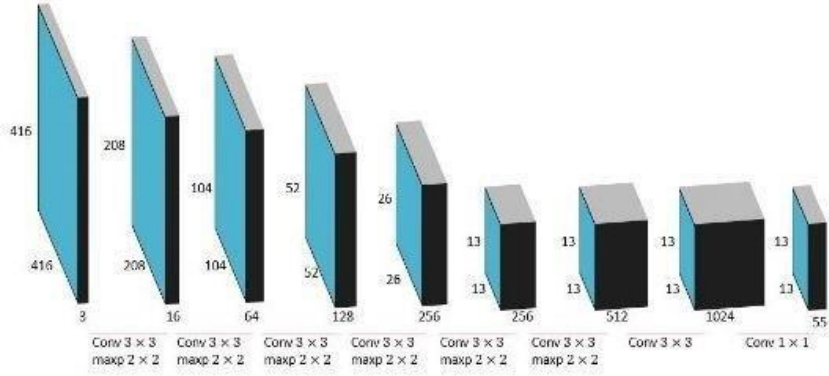
En hızlı şekilde gerçek zamanlı nesne tespiti yapabilen bir algoritma olarak 2017 yılında ortaya çıkmıştır. Diğer algoritmalara göre hızlı ve doğruluk oranı yüksek bir algoritmadır. Diğer algoritmalarda, gerçek zamanlı nesne tespiti yapmaktadır, fakat YOLO algoritmasının ortalama kesinlik değeri yüksektir. YOLO, nesne algılama için evrişimli sinir ağlarını kullanan bir algoritmadır. Tanıma algoritmalarına kıyasla, bir algılama algoritması yalnızca sınıf etiketlerini öngörmekle kalmaz, aynı zamanda nesnelerin konumlarını da algılar. YOLO algoritmasının diğer algoritmalarından farkı, evrişimli model kullanarak, veriyi daha hızlı işlemesidir. YOLO algoritmasında sıkça kullanılan, COCO veri setidir. Bu veri seti ile yapılan çalışmada, büyük nesnelerde %98 başarı oranı sergilerken, daha küçük nesnelerde başarı oranının %60 civarında olduğu ortaya konulmuştur. Derin öğrenme yöntemlerinden olan evrişimli ağ temelli model görüntü işleme de kullanılırken ve YOLO (You only look once) ve benzeri algoritmaların nesne tespiti yönteminde kullanıldığı gözlenmiştir



Şekil Error! No text of specified style in document..4. YOLOv3 diğer nesne tespiti algoritmaları ile kıyaslama

YOLO algoritmasının diğer algoritmalarından hızlı olmasının sebebi, girdinin sadece bir kez nöral ağdan geçerek, çıktının sınıfını ve koordinatlarını tahminlemesidir. [9]

YOLO algoritması, proje kapsamında bir mobil uygulaması üzerinde kullanıldığından dolayı mobil cihazların sınırlı donanımları göz önünde bulunduran uygun bir çözüm kullanılmalıdır. Bu problemi çözmek amacıyla uygulama içinde normal YOLOv2 algoritması yerine YOLOv2 Tiny modeli kullanılmaktadır.



Şekil Error! No text of specified style in document..5. YOLOv2 Tiny ağ yapısı

YOLOv2 Tiny, YOLOv1'den daha az parametreye sahiptir. Şekil 5.2.de gösterildiği gibi YOLOv2 Tiny'nin ağ yapısında 9 evrişim (convolution) katmanından ve 6 maksimum havuzlama katmanları bulunmaktadır. YOLOv2 Tiny modeli 80 farklı nesne tespiti yapma imkanına sahiptir. Aynı zamanda gerçek zamanlı (Real Time) uygulamaları için yaklaşık olarak ≈ 40 FPS ile çalışmaktadır ve mobil cihazların depolama alanları daha etkin bir şekilde kullanabilmek için diğer modeller ile mukayese edilerek ≈ 40 MB ile çok küçük boyuta sahiptir. Bunun için bu proje kapsamında YOLOv2 Tiny modeli kullanması en uygun nesne tespiti algoritması olarak görülmektedir.

5.2. Flutter Frameworku İçinde Yapay Zeka Kullanımı

2.2. Önerilen Sistem Analizi kısmında gösterildiği gibi nesne tespiti görevi yapay zeka teknolojileri kullanarak gerçekleştirilmektedir. Proje kapsamındaki mobil uygulaması geliştirmek için Flutter frameworku kullanılmıştır. Bunun için Flutter frameworku kullanarak nesne tespiti gerçekleştirmek projenin en önemli görevi olarak görülebilmektedir. Flutter frameworku içinde yapay zeka uygulamaları gerçekleştirmek hedefiyle 3 çözüm bulunmaktadır:

- **Firestore MLKit:** Google'ın Firestore hizmeti kullanan mobil uygulamaları Google tarafından geliştirilen ve tasarlanan bazı yapay zeka uygulamaların algoritmaları kullanmasına olanak tanımaktadır.
- **API:** Çoğunlukla sunucular ve web arayüzleri ile çalışırken ve yapay zeka modeli çok sık güncellenmesini beklerken, makine öğrenimini uygulamalara dahil etmenin yöntemlerinden biridir. Model geliştiriciler tarafından API hizmetlerine sararak ve web tabanlı sunucuları kullanarak barındırmaktadır.

- **Cihaz Üzerindeki Model (On-Device Model):** On-Device modeller, doğrudan kullanıcıların cihazlarında yüksek performans ile modeli kullanmaya ihtiyaç durumunda etkili bir yöntem olmaktadır.

Proje içerisinde Firebase hizmetleri kullanılmadığından dolayı Firebase MLKit çözümü uygun değildir. Aynı zamanda gerçek zaman olarak nesne tespiti yapılacağından API çözümü de istenilen performansı sağlaması mümkün olmadığından dolayı istenilen hızı ve performansı sağlayabilecek On-Device model çözümü kullanılmaktadır.

5.2.1. Flutter İçinde TensorFlow İle Eğitilen Model Kullanımı

Flutter içinde On-Device modelleri kullanmak amacıyla öncelikle model, TensorFlow ile oluşturulmakta ve eğitilmektedir. Eğitilen model .tflite dosyaları olarak export edilmektedir ve Flutter projenin assets klasöründe depolanmaktadır. Depolanan .tflite dosyası (modeli) tflite kütüphanesi kullanarak projede kullanılabilir hale gelmekte ve sonuçlarından faydalanmaktadır. Projede kullanılacak YOLOv2 Tiny modeli assets kısmında depolanmaktadır ve tflite kütüphanesi kullanarak recognition sınıfında modelin çıktılarını (tespit ettiği nesneleri) alınmakta ve projede kullanılabilir hale gelmektedir.

5.2.2. Tespit Edilen Nesneler Üzerinde İşlemler

YOLOv2 Tiny modelin çıktılarını recognition sınıfı ile işlenmektedir. Modelin en önemli çıktıları:

- **detectetClass:** YOLO modeli ile tespit edilen nesnelerin YOLO modelin içerdiği ve tanıdığı nesnelerin hangi sınıfına ait olduğunu ismini geri dödürerek göstermektedir.
- **confidenceInClass:** YOLO modeli ile tespit edilen nesnelerin doğruluk oranı (kesinlik derecesi) göstermektedir.
- **rect:** YOLO modeli ile tespit edilen nesnelerin işlenen görüntü (frame) üzerinde ekranın hangi kısmında bulunduğunu Bounding Box ismine sahip dikdörtgen ile gösterilmektedir. Nesnenin yerini gösteren dikdörtgenin konumu, uzunluğu ve genişliği içermektedir.

Modelin çıktısı olarak verilen bu parametreleri kullanarak proje kapsamındaki bulunan yapay zeka işlemleri gerçekleştirilmektedir.

5.2.2.1. Model Tarafından Tanınan Nesne Sınıfların İsimlerini

YOLOv2 Tiny modeli ile tespit edilen nesnelerin sınıflarının adlarını İngilizce dili ile tanınmaktadır. Ancak projenin potansiyel kullanıcıları Türkiye içinde yaşayan görme engellilerden oluşması sınıfların adlarını Türkçe dilinde olmalarını zorunlu hale getirmektedir. Bunun için tanınan nesnenin adını direk bir şekilde modelden alma yerine tanınan sınıfın sıralaması (index'ini) alınmakta ve model ile aynı klasörde bulunan nesnelerin Türkçe isimlerini içeren metin dosyası kullanarak tanınan nesnenin Türkçe sınıf adı model tarafından belirlenen indise göre alınmaktadır.

5.2.2.2. Bounding Box Çizimi

Modelin çıktısı olarak verilen bu parametrelerden **rect** parametresi kullanarak ekran üzerinde nesnelerin konumlarını kırmızı renkli dikdörtgen ile çerçevelenmektedir. rect parametrenin left ve top özellikleri ile dikdörtgenin görüntü üzerindeki konumunu belirlenmektedir. Aynı zamanda rect parametrenin width ve height özelliklerinde faydalanarak önceden konumunu belirlenen dikdörtgenin alanı verilmektedir. Bounding box'ların her birisi Flutter'ın Container Widget'i ile tasarlanmaktadır.



Şekil Error! No text of specified style in document..6. YOLOv2 Tiny modeli ile tespit edilen bir nesnenin Bounding Boxu

5.2.2.3. Uygulamada Tanınmayacak Nesneler

YOLOv2 Tiny modelinde tanınması mümkün olan nesnelerin sayısı 75'ten fazla fazladır. Ancak tanınan bu nesnelerin çoğu, projed görevi gerçekleştirmekte faydası olmayan nesnelerdir. Örneğin görme engellisi yürürken karşısına tekne yada zürafa çıkması mümkün olmayan yada çok düşük olasılığına sahip olduğundan tanınması beklenen nesnelerden kaldırılmaktadır. Bu nesnelerin isimlerinin nul şeklinde yazarak projenin içeriğine alındığı zaman if koşul bloku ile nul ismine sahip olan nesneleri işlenmemektedir.

5.3. Flutter Frameworku İçinde Metinden Yazıya Dönüştürme

2.2. Önerilen Sistem Analizi kısmında gösterildiği gibi yapay zeka modeli ile tespit edilen nesnelerin sınıflarının isimlerini görme engelliye karşısına gelebilecek nesnelerden uyarmak amacıyla sesli olarak söylenmektedir. Bunun gerçekleştirmek için 2 yol vardır:

- **FlutterTTS:** Flutter frameworku içinde metinden yazıya dönüştürme işlemi gerçekleştirmek için kullanılan bir kütüphanedir. Bu kütüphane her mobil cihazında bulunan metinden yazıya dönüştürme motoru kullanmaktadır. Android mobil cihazları için Google'ın ve IOS cihazları için Apple'in TTS motoru kullanılmaktadır. Bu motorlar 100'den fazla dil desteklemektedir.
- **Audio Dosyaları:** Bu proje kapsamındaki tanınması ve sesli olarak söylenmesi beklenmiş olan nesne sınıflarının isimlerini belli bir sayıda sınırlı olduğundan dolayı sınıfların isimlerini bir dosya halinde assets klasöründe depolayarak kullanılabilir.

Google/Apple TTS motoru online olarak sınırsız bir şekilde kullanılabilir. Ancak motoru offline ve ücretsiz olarak kullanmak için sınırlı sayıda kullanma hakkı verilmektedir ve bu ücretsiz kullanma hakları bitirdikten sonra her bir kullanma ücretli hale gelmektedir. Bunun için bu proje kapsamında gerçekleştirilecek görevi en iyi şekilde sağlamak için proje, offline ve sınırsız bir şekilde kullanılması gerekmektedir. Bu bilgilere istinaden audio dosyaları kullanımı daha etkin görülmektedir.

Depolanan audio dosyaların isimlendirmesi basit bir kurala göre yapılmaktadır. İsimlendirme *nesneinKonumunKısaltması + sınıfınİsmi* şeklinde yapılmaktadır. Konumların kısaltmaları **o** = **önünde**, **sg** = **sağında** ve **sl** = **solunda** şeklinde tasarlanmaktadır. Örneğin: "önünde insan var" cümlesine sahip olan audio dosyanın ismi **oinsan** şeklinde yazılmaktadır.

Depolanan bu audio dosyaları çalıştırmak için *assets_audio_player* kütüphanesi kullanılmaktadır.

Birden fazla nesne tespit edildiği zaman nesnelerin audio dosyaları iç içe bir şekilde çalışmasını engellemek amacıyla tespit edilen herbir nesnenin sınıfının ismini sesli olarak söylendikten sonra sonraki nesnenin ismini söylemek için 0,5 saniye beklenmektedir.

5.4. Flutter Frameworku İçinde Kullanıcı Arayüzü

Flutter frameworku kullanıcı arayüzü kiti (UI Kit) olduğundan dolayı kullanıcı arayüzleri tasarımı kolaylanmaktadır. Bu proje, görme engelliler tarafından kullanılacağından dolayı mümkün olduğu kadar arayüz sade ve etkileşimi en aza indirgeyen tasarım gerçekleştirmeye odaklanmaktadır. Şekil 5.4. gösterildiği gibi uygulamaya basıldığı zaman splash screen gösterilmekte ve ondan sonra buton gibi hiçbir tasarım elementi kullanmadan çalışmaya başlamaktadır. Kullanıcı arayüzü tasarlanırken görme engelliler tarafından uygulamanın kullanımı gözlere kolaylaştırma hedefiyle kalın ve siyaha yakın renkler kullanmakta özen gösterilmektedir.

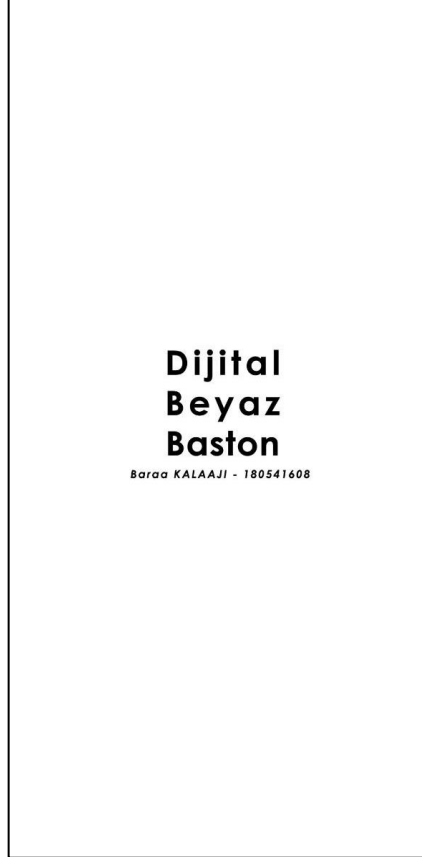


Şekil Error! No text of specified style in document..7. Kullanıcı Arayüzü

6. BULGULAR VE TARTIŞMA

6.1. Bulgular

Proje kapsamındaki yapılan uygulamayı çalıştırdıktan sonra splash screen ekranı gösterilmektedir.

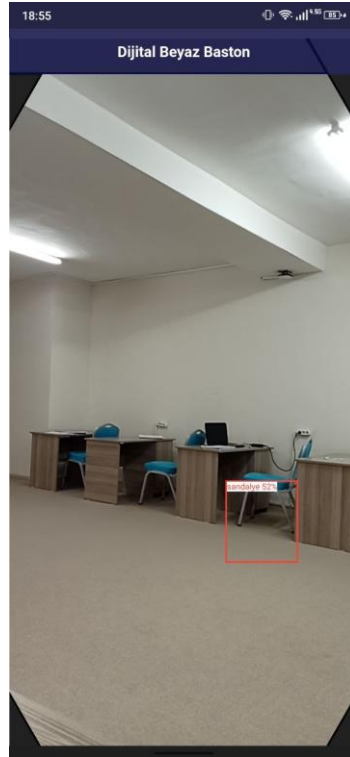


Şekil8. Kullanıcı Arayüzü

Splash screen süresi geçtikten sonra uygulama çalışmaya başlamaktadır. Projenin çalışması göstermek için audio dosyaların seslerini duymaya ihtiyaç bulunmaktadır. Ancak çalışmayı bu tez kapsamında göstermek için ekranın görüntüsü alınacak ve o anda çalışan audio dosyaları yazılacaktır.



Şekil.9. Çalışma Örneği - 1



Şekil.10. Çalışma Örneği - 2

6.2. Tartışma

Projenin gerekleřtiriminde analizi yapılan gereksinimleri gz nnde bulundurarak yapılmıřtır. Bunun iin splash screen tasarımına bakılırsa uygulamanın bařlangıcı az grmeye sahip kullanıcılara uyarmak hedefiyle beyaz rengi ile tasarlanmaıřtır.

rnek olarak alınan uygulamanın ıktılarında iki farklı ortam denenmiřtir. alıřma rneęi 1'deki ortam, uygulamanın gereksinim analizi kısmında yapılan analizlere gre beklenen ortamdır. Bunun iin grldę gibi rnek 1'de ekranın sol kısmında bisiklet ve orta kısmında insan bulunduęundan dolayı "*nnde insan var , solunda bisiklet var*" audio dosyayı alıřmıřtır. rnek 2'deki ortam beklenen ortamdaki farklı olmasında raęmen tanınması ve tespit edilmesi beklenen nesneleri doęru bir řekilde tanınmaktadır ve rnek 2 iin "*nnde sandalye var*" audio dosyası alıřmaktadır.

alıřan audio dosyaları sistemin grevi gerekleřtirmesine raęmen kısa bir zaman aralık iinde tekrarlandıęından dolayı kullanıcı iin rahatsızlık oluřturabilmektedir. Aynı zamanda kullanıcının mobil cihazının arka kamerası srekli olarak kullanımsız mobil cihazın pili hızlı bir řekilde bitmesine sebep olabilmekte ve mobilin donanım kaynakları tkenebilmektedir.

7. SONUÇLAR

Bu çalışma kapsamında yapılan görüldüğü gibi YOLOv2 Tiny modeli mobil uygulamalarında etkin ve performanslı bir şekilde çalışma kabiliyeti bulunmaktadır. Aynı zamanda YOLOv2 Tiny nesne tespiti modeli kullanarak görme engellileri yardım edebilecek yeni araçlar tasarlanabilmektedir. Bu proje kapsamında tasarlanan Dijital Beyaz Baston mobil uygulaması geleceğin tasarlanabilecek ve geliştirilebilecek benzer uygulamalara yol göstermektedir.

KAYNAKLAR

- [1] <https://blogs.worldbank.org/opendata/billion-people-experience-disabilities-worldwide-so-where-data>
- [2] <https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment>
- [3] <https://sbu.saglik.gov.tr/sbozurlu/>
- [4] TR52/18/TD/0039 -GÖRME ENGELLİLER SOSYO-EKONOMİK VE GİRİŞİMCİLİK DÜZEYİ ANALİZİ ARAŞTIRMASI
- [5] [https://tr.wikipedia.org/wiki/Dart_\(programlama_dili\)](https://tr.wikipedia.org/wiki/Dart_(programlama_dili))
- [6] [https://en.wikipedia.org/wiki/Flutter_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software))
- [7] <https://en.wikipedia.org/wiki/Flutter>
- [8] <https://tr.wikipedia.org/wiki/TensorFlow>
- [9] Tüfav Bilim Dergisi, cilt.13, sa.3, ss.1-14, 2020 (Diğer Kurumların Hakemli Dergileri)
- [10] G. Li, Z. Song and Q. Fu, "A New Method of Image Detection for Small Datasets under the Framework of YOLO Network," 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2018, pp. 1031-1035, doi: 10.1109/IAEAC.2018.8577214.