

**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
5. MATERYAL VE METOT	8
6. BULGULAR VE TARTIŞMA.....	10
7. SONUÇLAR.....	13
KAYNAKLAR.....	14

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 3.1. Sistem Mimarisi	5
Şekil 3.2. Sistem Mimarisi – Akış Şeması	5
Şekil 3.3. Kullanıcı Arabirimi - 1 ve 2 ve 3	6
Şekil 5.1. Kullanıcı Arayüzü - 1	10
Şekil 5.2. Kullanıcı Arayüzü - 2	10
Şekil 5.3. Kullanıcı Arayüzü - 3	10

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 yaklaşılsa 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 incelemesinin 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 görülmektedir. Bunun için bu çalışma kapsamında önerilen sistemin analizini yaparak mevcut sistemdeki bulunan dezavantajları 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 Play Store ve App Store uygulamalarında tüm özellikleri ile ücretsiz olmasından, kullanıcılar uygulamayı kolayca indirebileceklerdir.

Önerilen sistem, mevcut olan sistemler ile mukayese edildiği zaman, büyük veri seti ile eğitilen bir yapay zekâ modeli kullandığından yakın performansa sahip olması ile, ücretsiz olduğundan daha erişilebilir olması ile ve geliştirilen kamera uygulamasının ayarları ile farklı ışık ortamlarında kolayca kullanılabilir olması ile önerilen sistem, mevcut sistemdeki bulunan dezavantajları ortadan kaldırdığını göstermektedir.

3. SİSTEM TASARIMI

3.1. Genel Tasarım Bilgileri

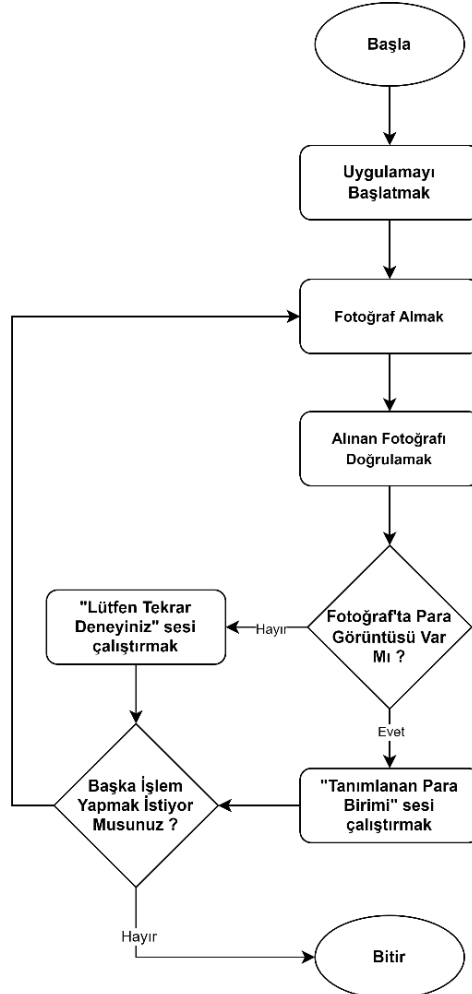
3.1.1. İşlevsel Belirtiler

İşlevsel belirtilerin tasarımı, sistemin ve yazılımın temel görevini tanımlayarak belirlenmektedir. Sistem, az görme engellilerine yardımcı uygulama geliştirmeye hedeflemektedir. Uygulama, TensorFlow Lite kütüphanesi ile eğitilen para tanıma modeli kullanmaktadır. Eğitilen yapay zekâ modeli Flutter framework’u kullanarak implementasyonu yapılacaktır.

3.1.2. Sistem Mimarisi



Şekil.3.1 Sistem Mimarisi



Şekil.3.2 Sistem Mimarisi – Akış Şeması

3.1.3. Testler

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

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

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

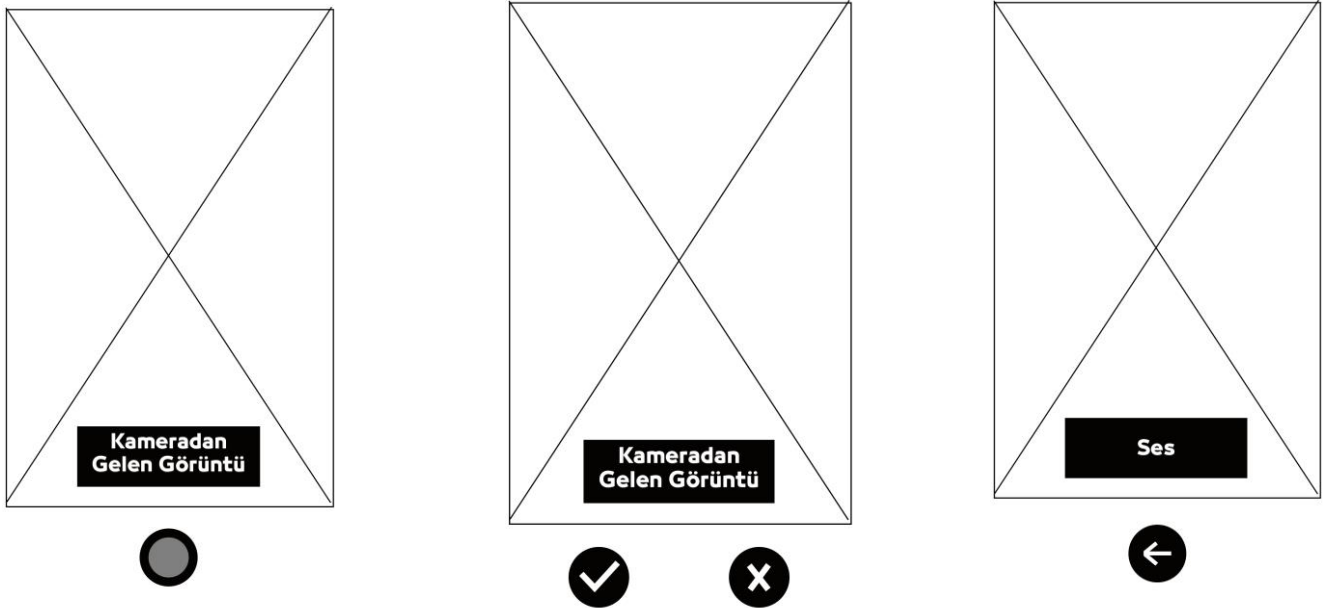
3.2. Süreç Tasarımı

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.3.3

Kullanıcı Arabirimi - 1 ve 2 ve 3

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]

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.[7]

Mobil uygulamanın gerçekleştirmesinde Dart programlama dili kullanan Flutter frameworku kullanılmaktadır. Flutter, çok iyi dokümantasyona 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. Üç Katmanlı Mimari (Three-Tier Architecture)

Üç katmanlı mimari, sunum katmanı, uygulama katmanı ve veri katmanından oluşan modüler bir istemci-sunucu mimarisidir. Veri Katmanı (Database Tier) bilgileri depolar, Uygulama Katmanı (Business Tier) mantığı işler ve Sunum Katmanı (Presentation Layer) diğer iki katman ile iletişim kuran bir grafik kullanıcı arabirimidir (GUI). Üç katmanlı mimari, işlevlerin yapılarını ayrılaştırarak performans ve güvenlik arttırabilen mantıksal bir yapıdır. Üç katmanlı mimari, projenin herbir kısmı güvenli bir şekilde oluşturabilmek amacıyla kodun tasarımında kullanılmıştır.

4.2.2. 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.

4.2.3. 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.

4.2.4. 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 zekâ modeli kullanıldığı kısımdır. Bu kısım içinde yapay zekâ modelin ürettiği çıktıları elde edilmekte ve kullanıma hazır olmaktadır.

- **Pages:** Uygulama içinde kullanılan farklı sayfaların tasarımının yapıldığı kısımdır. Bu kısım içinde her bir sayfanın widget yapısı bulunmaktadır.
- **Services:** TensorFlow ve audio dosyaları çalıştırma (Assets Audio Player) kütüphaneleri gibi kullanıldığı kısımdır.

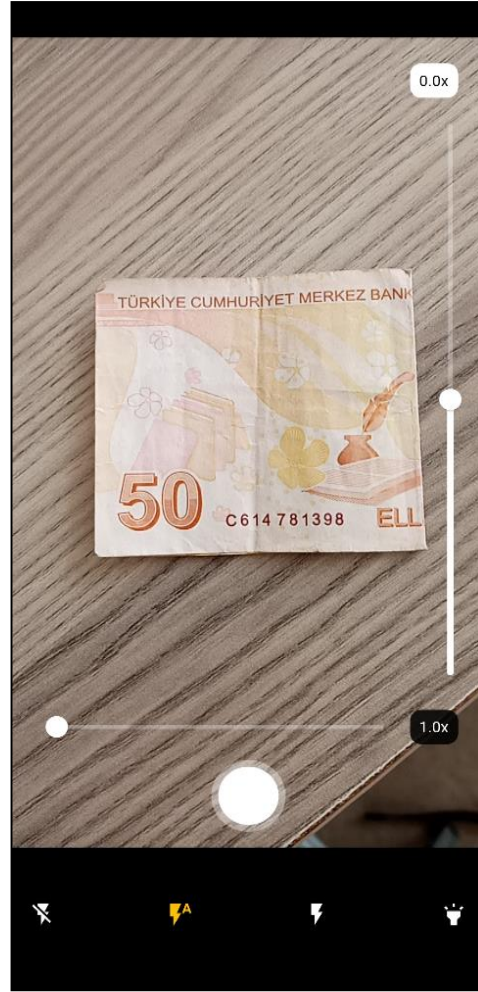
4.3. Olağandışı Durum Tanımları

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. BULGULAR VE TARTIŞMA

5.1. Bulgular

Projede yapılan uygulamayı çalıştırdıktan sonra para görüntüsü alabilmek için kamera uygulamasının ekranı gösterilmektedir.



Şekil.5.1 Kullanıcı Arayüzü - 1

Kullanıcı, parayı kamera görme açısı içinde koyduğu zaman ve daire şeklindeki capture butonuna bastığı zaman kamera vasıtasıyla görüntü alınacak ve doğrulama ekranına yönlendirilecektir. Doğrulama ekranındaki tick işaretine basıldıktan sonra tanınmış olan para birimi sesli olarak söylenecektir. Projenin çalışması bu tez kapsamında göstermek için ekranın görüntüsü alınacak ve kullanıcı arayüzün (UI) tüm ekranları gösterilecektir.



Şekil.5.2 Kullanıcı Arayüzü - 2



Şekil.5.3 Kullanıcı Arayüzü – 3

5.2. Tartışma

Projenin gerçekleştiriminde analizi yapılan gereksinimleri göz önünde bulundurarak yapılmıştır. Bunun için kamera ve uygulamanın diğer arayüzlerin tasarımına bakılırsa, uygulamanın kullanıcıları az görme engelliklerine sahip olan insanlar tarafından kullanılacağı göz önünde bulundurarak tasarlanmıştır.

Proje kapsamında geliştirilmiş ve kullanılmış olan TensorFlow Lite modeli kullanan kod parçacıkları düzenli bir sınıf şeklinde yaparak kolay yeniden kullanılabilen yapı şeklinde tasarlanmış olmaktadır.

Tasarlanmış bu sınıflar Flutter framework'u içinde gerçekleştirilmesi istenilen herhangi bir yapay zekâ uygulaması için yer alabilmektedir.

Proje kapsamında tasarlanan kamera uygulaması ve yapay zekâ kullanma sınıfı Flutter içinde kullanılmaya hazırlanmıştır.

6. SONUÇLAR

Bu çalışma kapsamında yapılan uygulamada görüldüğü gibi TensorFlow Lite Kütüphanesi ile eğitilen yapay zekâ modeli mobil uygulamalarında etkin ve performanslı bir şekilde çalışma kabiliyeti bulunmaktadır. Aynı zamanda özel veri seti ile eğitilen para tanıma yapay zekâ modeli kullanarak görme engellileri yardım edebilecek yeni araçlar tasarlanabilmektedir. Bu proje kapsamında tasarlanan Türk Lirası Para Tanıma mobil uygulaması geleceğin tasarlanabilecek ve geliştirilebilecek benzer uygulamalara yol çizmektedir.

KAYNAKLAR

- [1] <https://blogs.worldbank.org/opendata/billion-people-experience-disabilities-worldwide-so-where-s-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://tr.wikipedia.org/wiki/TensorFlow>