

YAPAY ZEKA DESTEKLİ GÖZ HASTALIKLARININ TEŞHİSİNDE MOBİL UYGULAMA GELİŞTİRİLMESİ VE RANDEVU SİSTEMİ ENTEGRASYONU - GÖZ ASİSTANI

ÖZET

Bu çalışmada, göz fundus görüntüleri üzerinden diyabetik retinopati, glokom, katarakt ve normal göz durumlarını otomatik olarak teşhis eden ve teşhis sonucuna göre hastanın konumuna en yakın uzman doktora randevu alabileceği entegre bir sistem geliştirilmiştir. Sistem; Flutter ile geliştirilmiş mobil uygulama, Spring Boot ile oluşturulmuş backend API, Python Flask ile entegre edilmiş yapay zeka modeli ve PostgreSQL veritabanı bileşenlerinden oluşmaktadır. MobileNetV2 tabanlı derin öğrenme modeli, 4000'den fazla göz fundus görüntüsü üzerinde eğitilerek %84.1 genel doğruluk ve 80.53 'lük test doğruluk oranına ulaşmıştır. Randevu sistemi, hastaların teşhis sonuçlarına göre uygun uzmanlık alanındaki doktorlara erişimini kolaylaştırmakta ve konum bazlı filtreleme yaparak en yakın sağlık kuruluşlarını önermektedir. Bu çalışma, göz hastalıklarının erken teşhisi, hastaların uygun uzmanlara yönlendirilmesi ve sağlık hizmetlerine erişimin artırılması konularında önemli katkılar sunmaktadır.

1. GİRİŞ

1.1. Problem Tanımı

Göz hastalıkları, dünya genelinde önemli bir sağlık sorunu olup, 2.2 milyardan fazla insanı etkilemektedir. Dünya Sağlık Örgütü (WHO) verilerine göre, dünya çapında 1 milyardan fazla insan, önlenemez veya henüz tedavi edilmemiş görme bozukluklarına sahiptir [1]. Özellikle diyabetik retinopati, glokom ve katarakt gibi yaygın göz hastalıkları, erken teşhis edildiğinde tedavi edilebilir veya ilerlemeleri yavaşlatılabilir olmasına rağmen, geç teşhis durumunda kalıcı görme kaybına yol açabilmektedir. Göz hastalıklarının teşhisinde yaşanan temel sorunlar şunlardır:

- Uzman Eksikliği ve Erişim Zorluğu
- Erken Teşhis Eksikliği
- Randevu Sistemlerinin Verimsizliği
- Konum Bazlı Eşleştirme Eksikliği

1.2. Mevcut Çalışmalar

Göz hastalıklarının teşhisinde yapay zeka uygulamaları konusunda literatürde önemli çalışmalar bulunmaktadır:

Gulshan ve arkadaşları (2016), diyabetik retinopati tespitinde GoogleNet Inception-v3 mimarisini kullanarak %87.4 AUC (Area Under Curve) değerine ulaşmıştır [4]. Ting ve arkadaşları (2017), diyabetik retinopati, olası glokom ve AMD (Yaşa Bağlı Makula Dejenerasyonu) için geliştirdikleri derin öğrenme sisteminde sırasıyla %94.1, %96.4 ve %93.2 AUC değerleri bildirmişlerdir [5]. Li ve arkadaşları (2019), VGG-16 mimarisini kullanarak glokom tespitinde %94.7 doğruluk oranına ulaşmıştır [6].

Bununla birlikte, mevcut çalışmaların çoğu şu eksiklikleri barındırmaktadır:

1. Çalışmalar genellikle laboratuvar ortamında ve sınırlı veri setleri üzerinde test edilmiştir.
2. Teşhis sonrası hasta yönlendirme ve takip sistemleri entegre edilmemiştir.
3. Mobil cihazlara uyarlanabilir, hafif model mimarileri yeterince incelenmemiştir.
4. Sınıflandırma genellikle tek bir göz hastalığına odaklanmıştır (çoğunlukla diyabetik retinopati).
5. Kullanıcı deneyimi ve uygulama kullanılabilirliği yeterince dikkate alınmamıştır.

1.3. Çalışmanın Amacı ve Katkısı

Bu çalışmanın temel amacı, göz hastalıklarının erken teşhisini kolaylaştıracak, hastaları uygun uzmanlara yönlendirecek ve sağlık hizmetlerine erişimi artıracak kapsamlı bir sistem geliştirmektir. Çalışmamızın literatüre ve pratiğe katkıları şunlardır:

1. **Çoklu Hastalık Teşhisi:** Dört farklı göz durumunu (normal, katarakt, diyabetik retinopati ve glokom) tek bir model ile sınıflandırabilen derin öğrenme sistemi.
2. **Tam Entegre Çözüm:** Teşhisten randevu almaya kadar tüm süreci kapsayan end-to-end bir çözüm.
3. **Mobil Uyumlu Hafif Model:** MobileNetV2 tabanlı, mobil cihazlarda yüksek performansla çalışabilen model mimarisi.
4. **Konum Bazlı Doktor Önerisi:** Kullanıcının konumunu ve teşhis sonucunu birlikte değerlendirerek en uygun uzman doktor önerisi sunan algoritma.
5. **Kullanıcı Odaklı Tasarım:** Teknik bilgisi olmayan kullanıcıların da kolayca kullanabileceği bir arayüz tasarımı.
6. **Full-Stack Entegrasyon:** Flutter (mobil), Spring Boot (backend), Python Flask (model API) ve PostgreSQL (veritabanı) teknolojilerinin entegre edildiği kapsamlı mimari.

2. YÖNTEM

2.1. Sistem Mimarisi

Geliştirilen sistem, dört ana bileşenden oluşan katmanlı bir mimariye sahiptir:

2.1.1. Mobil Uygulama Katmanı

Kullanıcı arayüzü Flutter framework kullanılarak geliştirilmiştir. Bu katman aşağıdaki işlevleri yerine getirmektedir:

- Kullanıcı kaydı ve kimlik doğrulama
- Fundus görüntüsü yükleme
- Konum bilgisi alma ve coğrafi verileri işleme
- Teşhis sonuçlarını görüntüleme
- Doktor seçimi ve randevu yönetimi
- Randevu seçme ve iptal etme

2.1.2. Backend API Katmanı

Spring Boot framework (v3.4.4) kullanılarak geliştirilen bu katman, RESTful API servisleri sunmaktadır. Spring Security ve JWT token kullanılarak güvenli bir kimlik doğrulama sistemi uygulanmıştır. Backend katmanı şu temel bileşenlerden oluşmaktadır:

- **Controller Sınıfları:**
 - AuthController: Kullanıcı kaydı ve girişi işlemleri
 - DiagnosisController: Teşhis oluşturma ve görüntüleme
 - DoctorController: Doktor bilgileri yönetimi
 - AppointmentController: Randevu oluşturma ve yönetimi
- **Model ve Repository Katmanı:** Spring Data JPA kullanılarak PostgreSQL veritabanı ile iletişim sağlar.
- **Güvenlik Katmanı:** JWT tabanlı kimlik doğrulama ve rol tabanlı erişim kontrolü (RBAC) uygular.

2.1.3. Model API Katmanı

Python Flask framework kullanılarak geliştirilen bu katman, TensorFlow ile eğitilmiş yapay zeka modelini barındırmaktadır. API, HTTP istekleri aracılığıyla modele erişim sağlar ve tahmin sonuçlarını Backend API'ye iletir. Temel özellikleri:

- Görüntü ön işleme fonksiyonları
- Model yükleme ve tahmin işlemleri
- Tahmin sonuçlarının JSON formatında döndürülmesi

2.1.4. Veri Katmanı

PostgreSQL veritabanı kullanılarak veri kalıcılığı sağlanmıştır. Veritabanı şeması, aşağıdaki temel tabloları içermektedir:

- **Users:** Kullanıcı bilgileri
- **Diagnoses:** Teşhis sonuçları ve görüntü verileri
- **Doctors:** Doktor bilgileri ve uzmanlık alanları
- **Hospitals:** Hastane bilgileri ve konum verileri
- **Appointments:** Randevu bilgileri

2.2. Veri Seti ve Ön İşleme

2.2.1. Veri Seti Özellikleri

Çalışmada, toplam 4000'den fazla fundus görüntüsü içeren kapsamlı bir veri seti kullanılmıştır. Veri seti, dört farklı kategoride görüntüler içermektedir:

1. **Normal Fundus:** Herhangi bir patoloji içermeyen sağlıklı göz fundus görüntüleri
2. **Katarakt:** Lens opasifikasyonu gösteren fundus görüntüleri
3. **Diyabetik Retinopati:** Mikroanevrizmalar, hemoraji ve eksüdalar içeren fundus görüntüleri
4. **Glokom:** Optik disk çukurlaşması gösteren fundus görüntüleri

Veri seti, aşağıdaki oranlarda bölünmüştür:

- Eğitim seti: %70 (2800+ görüntü)
- Doğrulama seti: %15 (600+ görüntü)
- Test seti: %15 (600+ görüntü)

2.2.2. Veri Ön İşleme

Tüm görüntüler, model eğitimi öncesinde şu ön işleme adımlarından geçirilmiştir:

1. **Yeniden Boyutlandırma:** Tüm görüntüler 256x256 piksel boyutuna standardize edilmiştir.
2. **Normalizasyon:** Piksel değerleri $[-1, 1]$ aralığına normalize edilmiştir (MobileNetV2 için).

3. **Veri Artırma (Data Augmentation):** Eğitim veri setinde çeşitliliği artırmak için aşağıdaki teknikler uygulanmıştır:

- Rastgele döndürme (± 20 derece)
- Genişlik ve yükseklik kaydırma ($\pm \%20$)
- Yatay çevirme
- Zoom ($\pm \%20$)

2.3 Model Geliştirme

Transfer learning yaklaşımı kullanılarak MobileNetV2 tabanlı bir model geliştirilmiştir:

Model, Adam optimizer ve sparse categorical crossentropy kayıp fonksiyonu kullanılarak 20 epoch boyunca eğitilmiştir.

2.4. Model API ve Backend Entegrasyonu

Model API, Flask framework ile geliştirilmiş ve model tahminlerini JSON formatında dönecek şekilde yapılandırılmıştır. Backend API, bu tahminleri alarak uygun doktor önerilerini hesaplamakta ve kullanıcıya sunmaktadır. Entegrasyon sürecinde:

1. Görüntü verisi, mobil uygulamadan backend API'ye iletilir
2. Backend API, görüntüyü Model API'ye gönderir
3. Model API, görüntüyü işler ve tahmin sonuçlarını döndürür
4. Backend API, tahmin sonuçlarına göre uygun doktorları hesaplar
5. Sonuçlar mobil uygulamaya iletilir

2.5. Mobil Uygulama Geliştirme

Mobil uygulama, Flutter framework kullanılarak cross-platform bir yaklaşımla geliştirilmiştir. Ana özellikler:

- Kullanıcı kaydı ve kimlik doğrulama
- Göz fundus görüntüsü yükleme
- Teşhis sonuçlarını görüntüleme
- Konum bazlı doktor önerileri
- Randevu oluşturma ve yönetme

3. BULGULAR

3.1. Model Performansı ve Analizi

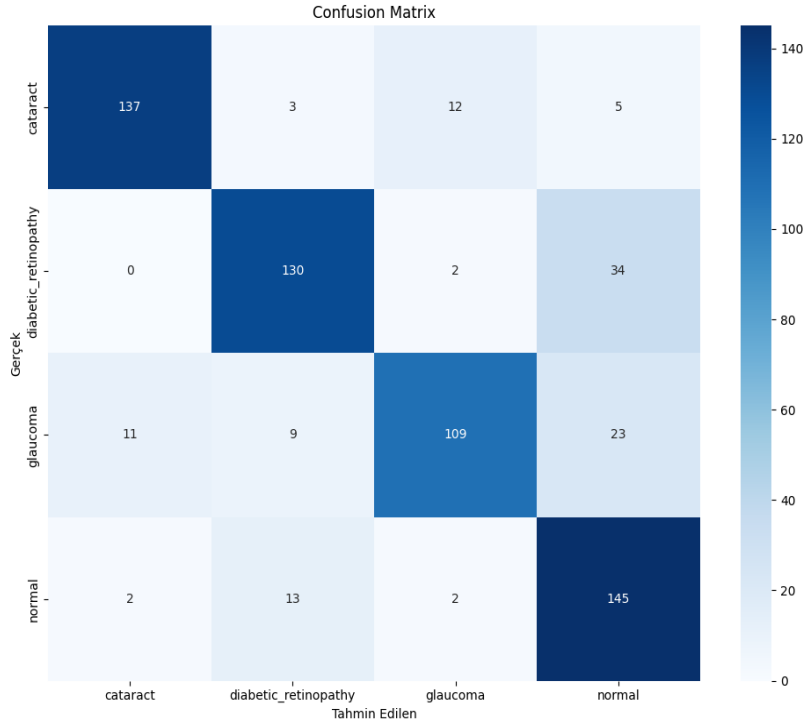
Eğitilen model, test veri setinde %84.1 genel doğruluk oranına ulaşmıştır. Sınıf bazlı performans metrikleri Tablo 1'de sunulmuştur.

Sınıf	Precision	Recall	F1-Score	Test Örnekleri
Normal	0.90	0.90	0.90	145
Katarakt	0.87	0.87	0.87	137
Diyabetik Retinopati	0.78	0.78	0.78	130
Glokom	0.71	0.72	0.71	109
Ağırlıklı Ortalama	0,84	0,84	0,84	0,84

Tablo 1: Sınıf Bazlı Performans Metrikleri

3.1.1. Confusion Matrix Analizi

Şekil 1'de görülen confusion matrix incelendiğinde, modelin özellikle normal (145/162) ve katarakt (137/157) sınıflarında yüksek doğruluk sergilediği görülmektedir. Diyabetik retinopati tespitinde 130 doğru tahmin yapılırken, 34 örnek normal olarak yanlış sınıflandırılmıştır. Glokom sınıfı ise 109 doğru tanı ile diğer sınıflara göre daha düşük performans göstermiştir.



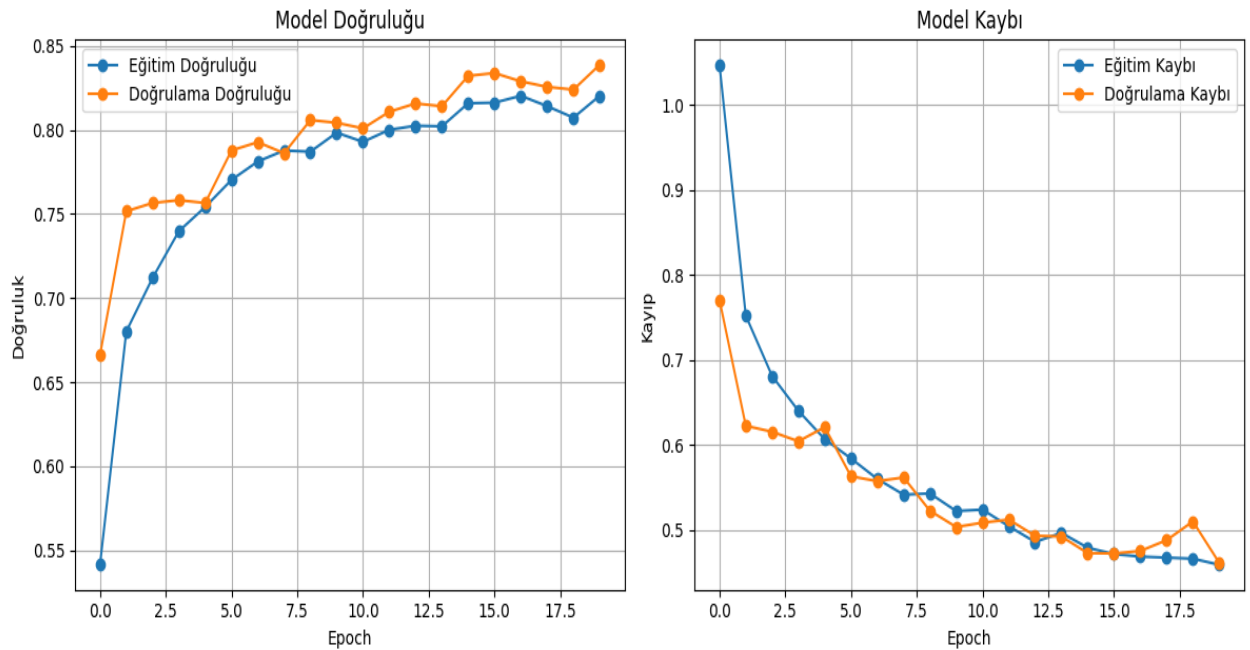
Şekil 1: Model Test Sonuçları Confusion Matrix

En önemli hata kaynakları şu şekilde tespit edilmiştir:

- Glokom örneklerinin 23 tanesi normal olarak yanlış sınıflandırılmıştır (%15)
- Diyabetik retinopati örneklerinin 34 tanesi normal olarak yanlış sınıflandırılmıştır (%20.5)
- Katarakt örneklerinin 12 tanesi glokom olarak yanlış sınıflandırılmıştır (%7.6)

3.1.2. Eğitim ve Doğrulama Süreçleri

Model eğitim sürecinde, Şekil 2'te görüldüğü üzere, eğitim ve doğrulama doğruluk oranları 18 epoch sonunda sırasıyla %81 ve %84'e ulaşmıştır. İlk epoch'ta %54 olan eğitim doğruluğu, 5. epoch'ta %76'ya, 10. epoch'ta %79'a yükselmiştir. Başlangıçta yüksek olan kayıp değerleri (1.05), eğitim ilerledikçe düşmüş ve 18. epoch'ta 0.47'ye kadar inmiştir.



Şekil 2: Model Eğitim ve Doğrulama Metrikleri

Epoch 10'dan sonra doğrulama doğruluğunun eğitim doğruluğunu geçmesi, modelin iyi genelleme yaptığını ve aşırı uyum (overfitting) sorunu yaşamadığını göstermektedir. Early stopping callback sayesinde doğrulama kaybının artmaya başladığı noktada eğitim durdurulmuştur.

3.2. Mobil Uygulama Özellikleri ve Kullanıcı Arayüzü

Geliştirilen mobil uygulama, kullanıcılara kapsamlı bir teşhis ve randevu deneyimi sunmaktadır. Ana ekranlar şunlardır:

1. **Giriş ve Karşılama Ekranı:** Kullanıcı kimlik doğrulama ve konum bilgisi görüntüleme
2. **AI Teşhis Ekranı:** Fundus görüntüsü yükleme alanı
3. **Teşhis Sonuç Ekranı:** Teşhis sonucu, güven skoru ve diğer olasılıklar
4. **Doktor Önerisi Ekranı:** Teşhis sonucuna ve konuma göre önerilen uzman doktorlar
5. **Randevu Ekranı:** Tarih ve saat seçimi, randevu oluşturma
6. **Randevu Listesi Ekranı:** Mevcut ve geçmiş randevuları görüntüleme

Uygulamanın temel özellikleri:

- **Konum Bazlı Hizmet:** Örneğin Battalgazi, Malatya gibi kullanıcının konumuna göre en yakın hastaneleri önerir
- **Teşhis Sonucu Görselleştirme:** Yapay zeka modelinin tahmin ettiği hastalığı ve güven skorunu gösterir
- **Detaylı Doktor Bilgileri:** Dr. Mehmet Demir gibi önerilen doktorların uzmanlık alanları ve iletişim bilgilerini gösterir
- **Esnek Randevu Sistemi:** 09 Mayıs 2025 gibi randevu tarihleri ve müsait saatler arasından seçim yapma imkanı
- **Randevu Takibi:** Mevcut randevuları görüntüleme ve iptal etme özelliği

5. SONUÇ

Bu çalışmada geliştirilen sistem, yapay zeka destekli göz hastalığı teşhisi ve randevu yönetimini entegre ederek göz sağlığı alanında önemli bir eksikliği gidermektedir. MobileNetV2 tabanlı model, %84.1 genel doğruluk, 80.53 test doğruluk oranıyla dört farklı göz durumunu başarıyla sınıflandırabilmektedir.

Sistemin konum bazlı doktor önerisi ve kullanıcı dostu arayüzü, özellikle göz sağlığı uzmanlarına erişimin sınırlı olduğu bölgelerde yaşayan bireyler için erken teşhis ve tedavi imkanı sunmaktadır.

Sonuç olarak, bu çalışma, yapay zeka, mobil sağlık uygulamaları ve sağlık hizmetleri entegrasyonu alanında yenilikçi bir yaklaşım sunarak, göz hastalıklarının erken teşhisine ve tedavisine katkıda bulunmaktadır.

KAYNAKLAR

- [1] WHO. (2021). World report on vision. World Health Organization.
- [2] Gulshan, V., et al. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA, 316(22), 2402-2410.
- [3] Ting, D. S. W., et al. (2017). Development and validation of a deep learning system for diabetic retinopathy and related eye diseases. JAMA, 318(22), 2211-2223.
- [4] Li, Z., et al. (2019). A Transfer Learning Approach for Automated Segmentation of Retinal Blood Vessels from Fundus Images. IEEE Transactions on Image Processing, 28(9), 4348-4358.
- [5] Howard, A. G., et al. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.
- [6] Sandler, M., et al. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4510-4520.
- [7] Russakovsky, O., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211-252.
- [8] Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359.