**Yüz Tanıma ile Bilgisayar Kilidi Açma Sistemi - Teknik Rapor**

**1. Giriş**

Bu projede, bilgisayar kamerası aracılığıyla kullanıcıların yüz görüntüleri üzerinden kimlik doğrulaması yaparak sisteme erişim sağlayan kamera tabanlı bir yüz tanıma ve kilit açma sistemi geliştirilmiştir. Sistem, kullanıcı kaydı, model eğitimi ve gerçek zamanlı yüz tanıma işlevlerini içeren bütüncül bir yapıya sahiptir. Grafiksel kullanıcı arayüzü (GUI), PyQt5 kütüphanesi kullanılarak tasarlanmış; yüz algılama ve tanıma işlemleri ise OpenCV kütüphanesi aracılığıyla gerçekleştirilmiştir. Tanıma sürecinde, özellikle düşük çözünürlükte ve değişen ışık koşullarında başarılı sonuçlar veren LBPH (Local Binary Patterns Histograms) algoritması kullanılmıştır. Proje, pratik kullanıma uygun, güvenli ve esnek bir biyometrik doğrulama altyapısı sunmayı amaçlamaktadır.

**2. Projenin Amacı**

Bu çalışmanın temel amacı, bireylerin yüz görüntülerine dayalı olarak kimlik doğrulamasını gerçek zamanlı ve kullanıcı dostu bir şekilde gerçekleştirebilecek bir biyometrik doğrulama sistemi geliştirmektir. Sistem, çoklu kullanıcı desteği sağlayarak farklı kişilerin yüzlerini öğrenebilmekte ve bu verileri kullanarak yalnızca tanımlı kullanıcıların erişimine izin vermektedir. Geliştirilen altyapı, kullanıcıların herhangi bir ek donanım gerektirmeksizin, yalnızca bilgisayara entegre bir kamera aracılığıyla kimliklerini doğrulayabilecekleri bir güvenlik çözümü sunmaktadır.

Yüz tanıma işlemleri sırasında kullanılan LBPH algoritması, düşük çözünürlüklü ve zorlu ışık koşullarında dahi tatmin edici sonuçlar verebilmesiyle öne çıkmaktadır. Bu algoritmanın tercih edilme sebebi, hem eğitim sürecinin hem de tanıma sürecinin hızlı ve düşük işlem gücüyle gerçekleştirilebilmesidir. Öte yandan, sistemin PyQt5 tabanlı grafik arayüzü, kullanıcı etkileşimini kolaylaştırmakta ve kayıt, model eğitimi, giriş denemeleri gibi işlemleri sezgisel olarak yönetilebilir kılmaktadır. Sonuç olarak bu proje, kaynak verimliliği yüksek, pratik kullanım odaklı ve güvenilir bir yüz tanıma tabanlı doğrulama sisteminin nasıl geliştirilebileceğine dair işlevsel bir örnek ortaya koymaktadır.

**3. Sistem Özellikleri**

* Yeni kullanıcı kaydı alma (kamera ile yüz görüntülerinin toplanması)
* LBPH algoritması ile model eğitimi
* Gerçek zamanlı yüz tanıma
* Eşik değerine göre güven skoruyla karar verme
* Tanınan/anonim girişleri log dosyasına yazma
* PyQt5 arayüzü ile kameradan canlı akış ve durum bilgisi gösterimi

**3.1 Kullanılan Teknolojiler**

* **Python**
* **OpenCV (opencv-contrib-python)**: Yüz tanıma için LBPH (Local Binary Pattern Histogram) algoritması kullanılmıştır.
* **PyQt5**: Grafiksel arayüz uygulaması için.
* **Haar Cascade**: Yüz tespiti için kullanılan önceden eğitilmiş algoritma.
* **NumPy**: Veri işleme için.
* **CSV**: Log ve etiket haritalarını tutmak için.

**4. Sistem Mimarisi ve Kod Yapısı**

**4.1 Kullanıcı Kaydı**

Sisteme yeni bir kullanıcı eklemek için, collect\_user\_data() fonksiyonu ile kullanıcıdan alınan metin girdisi (username) ile birlikte ./faces/{username} adlı bir klasör oluşturulur. Kamera aktif hâle getirilerek kullanıcıdan yüz görüntüleri alınır. Görüntüler üzerinde Haar Cascade sınıflandırıcısı kullanılarak yüz bölgesi tespit edilir. Algılanan her yüz, gri tonlamaya çevrilir, 200x200 boyutlarına yeniden boyutlandırılır ve cv2.imwrite() fonksiyonu aracılığıyla ilgili kullanıcı klasörüne .jpg formatında kaydedilir. Bu işlem, toplamda 100 farklı yüz görüntüsü elde edilene kadar devam eder.

**metin, insan yüzü, adam, insan, alın içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**  
Şekil 1. Kullanıcı yüzünden alınan bir frame ve counter gösterimi

**4.2 Eğitim**

load\_training\_data() fonksiyonu aracılığıyla ./faces/ dizini altındaki her bir kullanıcının kayıtlı yüz görüntüleri toplanır. Her kullanıcıya benzersiz bir etiket değeri (label\_id) atanır (örn. 0, 1, 2 ...). Toplanan tüm görüntü verileri ve karşılık gelen etiketler, OpenCV'nin cv2.face.LBPHFaceRecognizer\_create() metodu ile oluşturulan yüz tanıyıcı modele aktarılır ve model bu verilerle eğitilir. Eğitim tamamlandıktan sonra model parametreleri trained\_model.xml dosyasına kaydedilir. Buna ek olarak, kullanıcı adlarıyla etiket eşleşmelerini içeren label\_map.csv dosyası oluşturularak sistemin tanıma aşamasında bu eşleşmelerden yararlanması sağlanır.

metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 2. Label\_map.csv dosyasındaki kullanıcılar

**4.3 Gerçek Zamanlı Tanıma**

Uygulama başlatıldığında, eğitim aşamasında oluşturulan trained\_model.xml ve label\_map.csv dosyaları yüklenerek tanıma moduna geçilir. FaceRecognitionApp sınıfı üzerinden kamera aktifleştirilir ve her gelen kare (frame), update\_frame() fonksiyonu ile işlenir.

Her karede yüz tespiti yapılır ve yüz algılandığında:

* Yüz görüntüsü gri tonlamaya çevrilir ve normalize edilerek 200x200 boyutuna getirilir.
* model.predict() fonksiyonu yardımıyla tanımlanan yüzün etiket değeri (label) ve benzerlik skoru (confidence) hesaplanır.
* Elde edilen confidence değeri ile yüzün tanınma güveni şu formül ile yüzdeye çevrilir:  
  conf\_percent = int(100 \* (1 - confidence / 400))
* Eğer güven değeri önceden belirlenen CONFIDENCE\_THRESHOLD eşiğinin altındaysa (örneğin < 40), tanıma başarılı kabul edilir ve kullanıcı adı arayüzde görüntülenir.
* Tanıma sonucunda kullanıcı adı, zaman damgası, başarı durumu ve güven değeri login\_log.csv dosyasına kaydedilir.

**Not:** Başarılı tanımalarda çerçeve rengi yeşil, başarısız tanımalarda kırmızı olarak gösterilmektedir. Tanıma durumu ayrıca arayüzde "Unlocked" veya "Locked" şeklinde belirtilmektedir.

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

Şekil 3. login\_log.csv dosyasından bir görüntü

**5. Arayüz ve Kullanıcı Etkileşimi**

Uygulama açıldığında kullanıcıdan seçim yapması istenir:

* Yeni kullanıcı kaydı: Kamera açılır ve yüz görüntüsü toplanır.
* Tanıma moduna geçiş: Eğitilmiş model yüklenir, sistem tanıma moduna geçer.

**Arayüz Elemanları**:

* Görüntü ekranı (QLabel)
* Durum bilgisi (Status: Waiting..., Unlocked, Locked)
* Tanınan kullanıcı ismi ve doğruluk yüzdesi
* Gerçek zamanlı kare üzerine dikdörtgen ve yazı

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 4. Yeni kullanıcı kaydı arayüzü

Şekil 5. Tanıma moduna geçiş arayüzü

**6. Uygulama Sonuçları ve Test**

* Yüksek ışık koşullarında başarı oranı %90+ seviyelerine çıkmaktadır.
* Karanlık/arka ışıkta başarı düşmektedir.
* Eğitim sonrası yüz tanıma ortalama 0.2 saniye sürmektedir.
* mobil telefon, kişi, şahıs, metin, iç mekan içeren bir resim

  Yapay zeka tarafından oluşturulan içerik yanlış olabilir.Log dosyası ile tüm tanıma süreci izlenebilir hale getirilmiştir.

insan yüzü, ekran görüntüsü, adam, insan, kişi, şahıs içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.

Şekil 6. Yüz tanıma çıktısı(Locked)

Şekil 6. Yüz tanıma çıktısı(Unlocked)

**6.2. Güvenlik ve Hatalardan Koruma**

* Kullanıcı adlarındaki türkçe karakterler/boşluklar filtrelenir.
* Görüntü alınamazsa sistem uyarı verir.
* Kaydedilemeyen görüntüler konsola yazılır.

**7. Sonuç ve Değerlendirme**

Bu çalışma kapsamında, kamera tabanlı yüz tanıma teknolojisine dayanan, kullanıcı dostu bir kimlik doğrulama sistemi başarıyla geliştirilmiştir. Sistem, gerçek zamanlı olarak çalışabilmekte ve çoklu kullanıcı desteği ile genişletilebilir bir yapı sunmaktadır. Kullanıcı arayüzü aracılığıyla kullanıcı kaydı, model eğitimi ve giriş kontrolü gibi işlemler entegre biçimde yürütülmektedir. Bu sayede yalnızca yazılımsal bileşenlerle çalışan, harici sensör veya donanıma ihtiyaç duymadan kullanılabilen bir biyometrik güvenlik altyapısı oluşturulmuştur.

Yüz tanıma işlemlerinde kullanılan Local Binary Patterns Histograms (LBPH) algoritması, özellikle düşük donanım kaynaklarına sahip sistemlerde bile yüksek hızda çalışabilmesiyle öne çıkmaktadır. LBPH, yüzlerin yerel yapısal özelliklerini temel alan bir algoritma olduğundan, düşük çözünürlüklü görüntülerde ve sınırlı ışık koşullarında dahi tanıma performansı bakımından yeterli sonuçlar verebilmektedir.

Bununla birlikte, sistemin genel doğruluk seviyesini daha da artırmak ve farklı kullanıcıların birbirine benzer yüz yapılarından kaynaklanabilecek karışıklıkları önlemek amacıyla, gelecekte daha ileri düzey derin öğrenme tabanlı yöntemlerin entegre edilmesi önerilmektedir. Özellikle Dlib, FaceNet, ArcFace gibi embedding tabanlı modeller; kullanıcı yüzlerini vektör uzayında daha ayırt edilebilir şekilde temsil ederek daha yüksek tanıma doğruluğu sağlamaktadır. Bu tür algoritmalar, LBPH’ye kıyasla daha fazla işlem gücü gerektirse de, büyük veri kümelerinde ve kurumsal güvenlik sistemlerinde üstün performans göstermektedir.

Sonuç olarak bu proje, temel seviyede işlevsel ve yaygın olarak erişilebilir bir yüz tanıma sisteminin nasıl geliştirilebileceğine dair uygulanabilir ve genişletilebilir bir örnek sunmaktadır. Gerek teknik altyapısı gerek kullanıcı deneyimi açısından sağlam bir temel oluşturan bu sistem, gelecekte daha gelişmiş yapay zeka yaklaşımları ile desteklenerek daha kapsamlı uygulamalara dönüştürülebilir.

**8. Kaynakça**

[1] OpenCV, *OpenCV-Python Tutorials*, OpenCV.org, Accessed: May 2025.

[2] The Qt Company, *PyQt5 Documentation*, Qt for Python, Accessed: May 2025.

[3] OpenCV.org, “Face Recognition using Local Binary Pattern Histogram (LBPH),” *OpenCV Documentation*, Accessed: May 2025.

[4] GeeksforGeeks, “Face Recognition using Python and OpenCV,” *GeeksforGeeks*, Accessed: May 2025.

[5] OpenCV.org, “Face Detection using Haar Cascades,” *OpenCV Cascade Classifier Tutorial*, Accessed: May 2025.

[6] A. Rosebrock, “Face Recognition with OpenCV, Python and Deep Learning,” *PyImageSearch*, 2018. Accessed: May 2025.

[7] Python Software Foundation, *csv — CSV File Reading and Writing*, Python 3 Documentation, Accessed: May 2025.