T.C.
ERCİYES ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

İris Tanıma Sistemi

DANIŞMAN

Dr.Öğr.Üyesi ÖMÜR ŞAHİN

Bilgisayar Mühendisliği
YAZILIM MÜHENDİSLİĞİ DERSİ

FİNAL RAPORU

İRİS TANIMA SİSTEMİ

İris tanıma sistemi, bireylerin göz irislerini kullanarak kimliklerini doğrulamak veya tanımak için kullanılan bir biyometrik güvenlik teknolojisidir. Bu sistem, gözün renkli kısmı olan iris üzerindeki özel desenleri kullanarak bireyleri ayırt etmeyi sağlar. İris, gözün iç kısmında, öğrenci adı verilen açıklıklı bir alanın etrafında bulunur.

İris tanıma sistemi şu ana unsurlara dayanır:

1.İris Desen Tanıma:
İris, kişinin genetik yapısına özgüdür ve her bireyde benzersiz desenlere sahiptir. Bu desenler, damarlar, boşluklar ve diğer özellikleri içerir. İris tanıma sistemi, bu desenleri analiz eder ve kaydeder.

2.Kamera ile Görüntü Alma:
İris tanıma sistemi genellikle özel kameralar aracılığıyla gözün iris bölgesini yüksek çözünürlüklü görüntülerle yakalar. Bu kameralar, özel bir ışık spektrumu kullanarak iris desenlerini daha belirgin hale getirir.

3.Örüntü İşleme ve Şifreleme:
Elde edilen iris görüntüsü, örüntü işleme algoritmaları kullanılarak analiz edilir. Özel algoritmalar, iris desenini çıkarır ve bu deseni temsil eden bir matematiksel şifre oluşturur. Bu şifre, kişisel bilgileri içermediği için gizliliği korur.

4.Veritabanı Karşılaştırma:
İris tanıma sistemi, kaydedilmiş olan diğer iris desenleriyle karşılaştırma yapar. Bu desenler genellikle bir veritabanında saklanır. Bir kişi sisteme kayıt olurken oluşturulan iris şifresi, daha sonra tanıma işlemi sırasında karşılaştırılır.

5.Hızlı ve Güvenilir Tanıma:
İris tanıma sistemi, diğer biyometrik yöntemlere göre hızlı ve yüksek doğruluklu bir tanıma sağlar. İris deseni genetik olarak belirlendiği için değiştirilmesi veya taklit edilmesi zor olan bir biyometrik özelliktir.

İris tanıma sistemleri genellikle güvenlik sistemlerinde, bilgisayar sistemlerinde ve bazı mobil cihazlarda kullanılır. İris tabanlı kimlik doğrulama, parola veya kart bazlı sistemlere göre daha güvenli ve doğru sonuçlar sağlayabilir.

İris Tanıma Sistemi Nasıl Çalışır?

Bir iris tarama sistemini anlatma için, öncellikle gözünüzün benzersiz deseninin tanınması gerekir, böylece pozitif olarak tanımlanabilirsiniz. Bu, iris taramasında iki ayrı aşama olması gerektiği anlamına gelir: Kayıt (sistemi ilk kez kullandığımızda) ve doğrulama / tanıma. **1.KAYIT** ilk olarak, sistemin bilmesi gereken tüm insanlar gözlerini taramak zorundadır. Bu bir defalık işleme kayıt adı verilir. Her insan bir kameranın önünde durur ve gözleri hem sıradan ışık hem de görünmez kızılötesi ile dijital olarak fotoğraflanır. İris tanımada, kızılötesi, sıradan ışıkta net bir şekilde göze çarpmayan koyu renkli gözlerin benzersiz özelliklerini göstermeye yardımcı olur. Bu iki dijital fotoğraf daha sonra bir bilgisayar tarafından analiz edilir gereksiz ayrıntıları kaldırır (kirpikler gibi) ve yaklaşık 240 benzersiz özelliği tanımlar. Her göze özgü bu özellikler, bir bilgisayar veritabanında adınızın ve diğer ayrıntıların yanında ˇ saklanan İrisCode adı verilen 512 basamaklı basit bir sayıya dönüştürülür. Kayıt işlemi tamamen otomatiktir ve genellikle birkaç dakikadan fazla sürmez.

İris Tarama Kaydı Nasil Yapilir?

Bir biyometrik kimlik doğrulama biçimi olarak kullanmak ˇ için iris tanıma kaydında dört ana adım vardır: 1.Görüntü yakalama: Kişinin sol ve sağ irisinin yüksek kaliteli görüntüsü, özel bir ˇ iris kamera kullanılarak yakalanmalıdır. Bu kameralar, irisin dakika ve karmaşık ayrıntılarını, örneği kirletebilen görünür ışıktan (VIS) çok daha yüksek doğrulukla yakalamak için ˇ yakın kızılötesi (NIR) sensörler kullanır. Görünür ışık, bir öznenin gözlerine tutulduğunda rahatsızlık ve kasılmalara ne- ˇ den olabilir. 2.Uyumluluk kontrolü ve görüntü geliştirme: Bir sonraki adım, yakalanan görüntünün gelecekteki iris taraması için biyometrik ¸şablon olarak uygun olduğundan emin ˇ olmak için kalite ve uygunluk kontrolleri yapılmaktadır. Bu, her görüntüyü kaliteyi belirten, ancak bunlarla sınırlı olmamak üzere, anahtar özellikler için analiz eden özel bir yazılım gerektirir. -Keskinlik.

-Gri seviye yayılımı.

-Marj.

- İris sklera kontrastı.

- İris kontrastı ve pupiller dilatasyon.

-Kirpik varlığı.

-Göz kapağı tıkanıklığı.

İris Tanıma Sistemi Avantajları Ve Dezavantajları

Avantajlar:

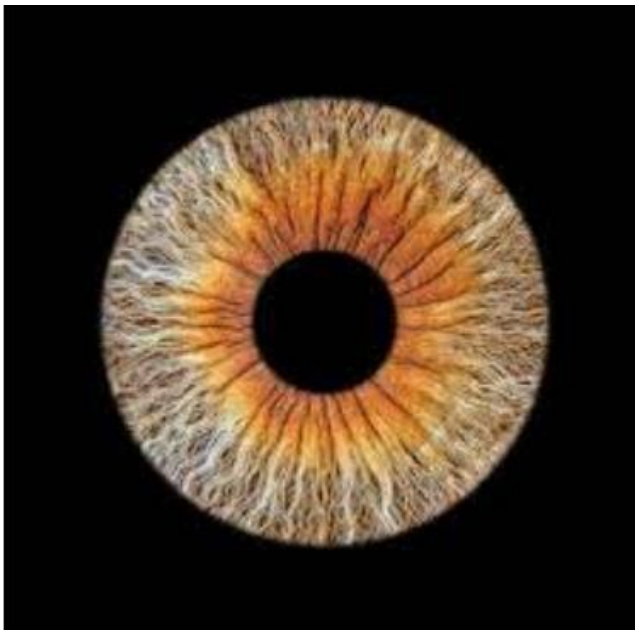
- **Yüksek Doğruluk:**
- İris deseni genetik olarak belirlendiği için kişinin yaşlanması veya dış etkenlere maruz kalması gibi durumlar, genellikle iris desenini etkilemez. Bu nedenle, iris tanıma sistemi diğer biyometrik sistemlere göre daha yüksek doğruluk sağlar.
- **Benzersizlik:**
- Her bireyin iris deseni benzersizdir. Bu, başka bir kişinin irisini taklit etmenin zor olması anlamına gelir, bu da sistem güvenliğini artırır.
- **Hızlı Tanıma:**
- İris tanıma, hızlı ve etkili bir tanıma sürecine sahiptir. İris deseni, bir kişinin gözüne bakılarak veya hızlı bir kamera taramasıyla anında elde edilebilir.
- **Dokunma Gerektirmez:**
- İris tanıma, kişinin gözünün kameralar tarafından yakalanmasıyla gerçekleşir. Bu nedenle, dokunma veya fiziksel temas gerekmez, bu da hijyenik bir çözüm sunar.
- **Gizliliğin Korunması:**
- İris tanıma sistemi, özel ve genetik bir özellik olan iris desenini kullanır. Bu desen, genellikle kişisel bilgileri içermez ve şifrelenmiş bir formda saklandığı için gizliliği korur.

Dezavantajlar:

- **Maliyet:**
- İris tanıma sistemleri genellikle diğer biyometrik teknolojilere göre daha pahalıdır. Hem donanım hem de yazılım maliyetleri yüksek olabilir.
- **Donanım Gereksinimleri:**
- Yüksek çözünürlüklü kameralar ve özel algoritmalar gerektirdiği için başlangıç maliyeti yüksektir. Bu, sistem kurulumunu ve sürdürülebilirliğini zorlaştırabilir.
- **Cooperasyon Gerektirmez:**
- İris tanıma, bireyin bilgisi veya izni olmadan uzaktan gerçekleştirilebilir. Bu, bireyin haberi olmadan tanıma işlemlerinin gerçekleştirilebileceği anlamına gelir.
- **Uzak Mesafe Zorlukları:**
- İris tanıma sistemi, genellikle belirli bir mesafeden gerçekleştirilir. Uzak mesafelerde, özellikle kişinin bilgisi olmadan tanıma yapma zorluğu vardır.
- **Gözlük veya Lens Kullanımı:**
- Bazı iris tanıma sistemleri, kişinin gözlük veya lens kullanması durumunda doğruluk oranlarını düşürebilir.
- **Toplumsal Kabul:**

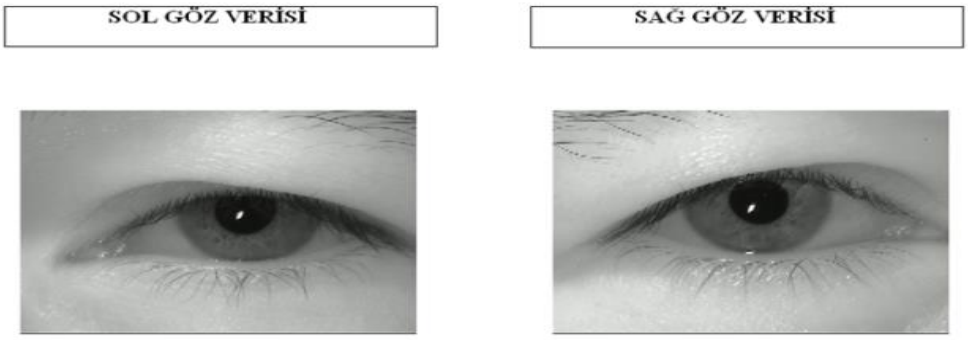
- Biyometrik tanıma teknolojileri genellikle toplumsal kabul konusunda çeşitli görüşlere neden olabilir. Bu, bazı kullanıcıların gizlilik endişeleri nedeniyle bu tür sistemlere karşı çıkmasına yol açabilir.

Her avantaj ve dezavantaj, uygulama bağlamına ve sistem tasarımına bağlı olarak değişebilir. Bu nedenle, bir iris tanıma sistemi uygulamadan önce dikkatlice planlanmalı ve değerlendirilmelidir.



Sekil 1: Sepete ekle butonları için yazılmış olan kod

Bu projede GoogleNet modeli kullanılarak özellik çıkarımı yapılmıştır. featureLayer degişkeni "loss3-classifier" olarak ˇ belirlenmiştir. Bu katman, GoogleNet'in çıktı katmanlarından biridir ve genellikle sınıflandırma için kullanılır. Bu projede, activations işlevi kullanılarak imdatastTrain ve imdatastTest veri kümelerindeki görüntüler GoogleNet'e verilerek featureLayer katmanının çıktıları alınmıştır. Bu çıktılar, eğitim ve test özelliklerini oluşturmak için kullanılmıştır. Daha sonra, fitcecoc fonksiyonu kullanılarak çok sınıflı bir sınıflandırıcı oluşturulmuştur. Bu sınıflandırıcı, eğitim özelliklerini ve etiketlerini kullanarak eğitilmiştir. Ardından, eğitilmiş sınıflandırıcı kullanılarak test özelliklerine tahminler yapılmıştır. Elde edilen tahminler, doğruluk hesaplaması için kullanılmıştır. best değişkeni, en iyi doğruluk değerini saklamak için kullanılmıştır. Her iterasyonda doğruluk de ˇ geri kontrol edilir ve daha yüksek bir doğruluk elde edilirse, sınıflandırıcı modeli model.mat dosyasına kaydedilir. Bu şekilde, en iyi doğruluk değerine sahip olan model kaydedilir.



Şekil 2: Veri setinde yer alan sag ve sol göze ait veriler

	1.KOŞMA	2.KOŞMA
Mean accuracy	0.846944	0.845889
Best accuracy	0.883333	0.872222

Şekil 3: Deney Tablosu

MATLAB Nedir?

MATLAB, Sayısal hesaplama, veri analizi, makine öğrenimi ve görselleştirme için kullanılan bir yazılım paketidir. Bilimsel ve mühendislik uygulamaları için güçlü bir araçtır.1970'lerde Massachusetts Teknoloji Enstitüsü'nde (MIT) geliştirildi ve o zamandan beri dünya çapında milyonlarca kullanıcı tarafından kullanılmaktadır.Aşağıdakiler dahil olmak üzere çok çeşitli bilimsel ve mühendislik uygulamaları için kullanılabilir:

Matematiksel hesaplamalar: MATLAB, karmaşık matematiksel hesaplamaları hızlı ve verimli bir şekilde gerçekleştirmek için kullanılabilir.

Örneğin, MATLAB, matris işlemleri, integraller, türevler ve diğer matematiksel fonksiyonları hesaplamak için kullanılabilir.

Veri analizi: MATLAB, veri analizinde güçlü araçlar sağlar. Örneğin, MATLAB, istatistiksel analiz, makine öğrenimi ve görselleştirme için kullanılabilir.

Makine öğrenimi: MATLAB, makine öğrenimi uygulamalarında yaygın olarak kullanılmaktadır. Örneğin, MATLAB, veri madenciliği, sınıflandırma ve regresyon için kullanılabilir.

Görselleştirme: MATLAB, veri görselleştirme için güçlü araçlar sağlar. Örneğin, MATLAB, grafikler, çizimler ve animasyonlar oluşturmak için kullanılabilir.

Simülasyon: MATLAB, simülasyonlar oluşturmak için yaygın olarak kullanılmaktadır. Örneğin, MATLAB, fiziksel sistemlerin, ekonomik modellerin ve diğer karmaşık sistemlerin simülasyonunu gerçekleştirmek için kullanılabilir.

MATLAB App Designer Nedir?

MATLAB App Designer, MATLAB'da etkileşimli kullanıcı arayüzleri (UI) oluşturmak için kullanılan bir araçtır. App Designer kullanarak, kullanıcıların etkileşimli bir şekilde etkileşime girebildiği ve verileri girdi olarak sağlayabildiği veya çıktı olarak alabileceği UI'lar oluşturabilirsiniz.MATLAB'ın UI oluşturma yeteneklerini genişletir. App Designer olmadan, UI'lar oluşturmak için MATLAB'ın yerleşik kodlama özelliklerini kullanmanız gerekir. Bu, kodlama deneyimine sahip kullanıcılar için uygun olsa da, kodlama deneyimine sahip olmayan kullanıcılar için zor olabilir.UI oluşturmayı kolay ve erişilebilir hale getirir. App Designer kullanarak, UI'ların görünümünü ve işlevselliğini sürükle ve bırak yöntemiyle tasarlayabilirsiniz. Bu, UI'ları daha hızlı ve kolay bir şekilde oluşturmanıza olanak tanır.

App Designer kullanarak oluşturabileceğiniz bazı UI örnekleri şunlardır:

Veri giriş ve çıktı formları: Veri giriş ve çıktı formları, kullanıcıların verileri MATLAB'a girmelerine veya MATLAB'dan verileri almalarına olanak tanır. Örneğin, bir veri giriş formu, kullanıcıların bir dizi sayı veya metni girmelerine olanak tanıyabilir.

Grafik kullanıcı arayüzü (GUI) araçları: Grafik kullanıcı arayüzü (GUI) araçları, kullanıcıların MATLAB'ın özelliklerini ve işlevlerini daha kolay bir şekilde kullanmalarına yardımcı olur. Örneğin, bir grafik araç çubuğu, kullanıcıların MATLAB'da grafikler oluşturmasına olanak tanıyabilir.

Oyunlar ve simülasyonlar: Oyunlar ve simülasyonlar, kullanıcıların etkileşimli bir şekilde etkileşime girebildiği ve mantıksal veya fiziksel sistemleri deneyimleyebildiği uygulamalardır. Örneğin, bir basit oyun, kullanıcıların bir topu bir çubuğa vurmasına olanak tanıyabilir.

MATLAB App Designer'ın Özellikleri -->

-Sürükle ve bırak tasarım: App Designer, UI'ların görünümünü ve işlevselliğini sürükle ve bırak yöntemiyle tasarlamanıza olanak tanır. Bu, UI'ları daha hızlı ve kolay bir şekilde oluşturmanıza olanak tanır.

-Otomatik kodlama: App Designer, UI'ların davranışını otomatik olarak kodlar. Bu, kodlama deneyimine sahip olmayan kullanıcılar için UI oluşturmayı kolaylaştırır.

-Kod düzenleme: App Designer, UI'ların davranışını düzenlemenize olanak tanıyan bir kod editörü içerir. Bu, kodlama deneyimine sahip kullanıcılar için UI oluşturmayı daha da özelleştirilebilir hale getirir.

-Test etme ve dağıtım: App Designer, UI'ları test etmenize ve dağıtmanıza olanak tanır. Bu, UI'larınızın düzgün çalıştığından ve kullanıcılarınız için erişilebilir olduğundan emin olmanıza yardımcı olur.

Sonuç olarak, bu özellikler, MATLAB'ı, UI oluşturma konusunda deneyimli kullanıcılar için de değerli bir araç haline getirir. Kod düzenleme özelliği, kullanıcıların UI'ların davranışını daha da özelleştirmelerine olanak tanır. Test etme ve dağıtım özellikleri, UI'ların düzgün çalıştığından ve kullanıcılarınız için erişilebilir olduğundan emin olmanıza yardımcı olur.

GoogleNet Nedir?

GoogleNet, Inception v1 olarak da bilinen bir derin öğrenme modelidir. GoogleNet, 2014 yılında Google tarafından geliştirilen ve ImageNet Large Scale Visual Recognition Challenge yarışmasında öne çıkan bir modeldir. Model, birçok paralel konvolüsyon katmanı içerir ve bu da öğrenme yeteneklerini artırır. GoogleNet, Inception blokları adı verilen özel mimarisiyle dikkat çeker.

Loss3 Classifier Nedir?

Loss3 terimi, genellikle bir derin öğrenme modelinin eğitimi sırasında kullanılan bir kayıp fonksiyonunu ifade eder. Derin öğrenme modelleri, eğitim sırasında gerçek etiketlerle tahmin edilen etiketler arasındaki farkı minimize etmeye çalışır. Kayıp fonksiyonu, bu farkı ölçen bir metrik olarak düşünülebilir.

Fitcecoc Fonksiyonu Nedir?

Fitcecoc terimi MATLAB'un Machine Learning Toolbox'unda kullanılan bir fonksiyondur. Bu fonksiyon, Çoklu Sınıf Desteği için Hata Düzeltme Çıkarma (Error-Correcting Output Codes - ECOC) yöntemini uygulamak için kullanılır. ECOC, çoklu sınıf sınıflandırma problemleri için bir stratejidir. Bu strateji, çoklu sınıf sorununu ikili (binary) sınıflandırma problemlerine dönüştürerek çözer.

Bu fonksiyon, verileri eğitmek için kullanılan özellik matrisini (features) ve sınıf etiketlerini alır. Ayrıca, kullanılacak ikili sınıflandırıcı türünü belirtmek ve modeli eğitmek için kullanılacak özellik seçeneklerini sağlamak gibi parametreleri içerir. **fitcecoc** fonksiyonu, çoklu sınıf sınıflandırma problemleri için çoklu ikili sınıflandırıcılar oluşturarak ve eğiterek çalışır.

Datasetimizde bulunan test ve train klasörlerinin içeriği nedir?

Train klasöründe kullanıcıdan alınan 280 adet sol göz verisi, 280 adet sağ göz verisi, test klasöründe ise kullanıcıdan alınan 120 adet sol göz verisi, 120 adet sağ göz verisi bulunmaktadır.

Alınan verilerin %70i train klasöründe, %30u test klasöründe bulunmaktadır. Datasetimiz bu bilgilerden oluşmaktadır.

Kodlar ve kodların detaylı açıklamaları:

```
classdef FolderOpener < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        UIFigure       matlab.ui.Figure
        GizleButton    matlab.ui.control.Button
        GozatButton    matlab.ui.control.Button
        TextArea       matlab.ui.control.TextArea
        TextAreaLabel  matlab.ui.control.Label
        TestButton     matlab.ui.control.Button
 end
    properties (Access = private)
```

```
    dataFolderTest  % Property to store the selected test data folder
    databaseFileName = 'hiddenDb';
    selectedFolderPath
    directoryToOpen = '';
end

methods (Access = private)
    function Iout = readandpreprocess(app, filename)
        I = imread(filename);
        if ismatrix(I)
            I = cat(3, I, I, I);
        end
        Iout = imresize(I, [224, 224]);
    end
end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: TestButton
    function TestButtonPushed(app, event)
        if isempty(app.dataFolderTest)
            app.TextArea.Value = [app.TextArea.Value; {'Lütfen veri dosyanızı seçin.'}];
            return;
        end


        loadedData = load(app.databaseFileName);
```

```
        dataFolderTrain = 'C:\Users\begum\Desktop\Projects\Yazılım Müh\Dataset\train';

        % dataFolderTest = 'C:\Users\begum\Desktop\Projects\Yazılım Müh\Dataset\test';

        categories = {'1', '2', '3', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23', '24', '25', '26', '27', '28', '29', '30', '31', '32', '33', '34', '35', '36', '37', '38', '39', '40', '41', '42', '43', '44', '45', '46'};


        imdatastTrain = imageDatastore(fullfile(dataFolderTrain, categories), "LabelSource", "foldernames");

        imdatastTrain.ReadFcn = @(filename)readandpreprocess(app,filename);

        imdatastTest = imageDatastore(fullfile(app.dataFolderTest, categories), "LabelSource", "foldernames");

        imdatastTest.ReadFcn = @(filename)readandpreprocess(app,filename);


        net = googlenet;


        featureLayer = 'loss3-classifie

        trainingFeatures = activations(net, imdatastTrain, featureLayer, "OutputAs", "columns");

        testingFeatures = activations(net, imdatastTest, featureLayer, "OutputAs", "columns");

        trainingLabels = imdatastTrain.Labels;

        testingLabels = imdatastTest.Labels;

        best = 0;

        for i = 1:3 % 3 iterasyon çalışacak

            classifier = fitcecoc(trainingFeatures, trainingLabels, "ObservationsIn", "columns", "Learners", "linear");


            predictedLabels = predict(classifier, testingFeatures, "ObservationsIn", "columns");

            accuracy = mean(predictedLabels == testingLabels);

            if accuracy > best
```

```
            best = accuracy;
            save('model.mat', 'classifier');
        end

        if best < 0.85
            app.TextArea.Value = [app.TextArea.Value; {'Erişim izniniz
bulunmamaktadır.'}];
            break;
        else
            app.TextArea.Value = [app.TextArea.Value; {sprintf('Iterasyon:%d
Doğruluk:%f', i, accuracy)}];
        end
    end

    if best >= 0.85
        if isfield(loadedData, 'hiddenData')
            hiddenData = loadedData.hiddenData
            app.TextArea.Value = [app.TextArea.Value; {sprintf('Girişe izin veriliyor. En
iyi doğruluk: %f\n', best)}];
            % .mat dosyasına kaydedilen dosya yolu
            hiddenFolderPath = hiddenData.folderPath;
            % Dosya yolunu winopen ile aç
            winopen(hiddenFolderPath);
        else
            app.TextArea.Value = [app.TextArea.Value; {'Aradığınız klasör
veritabanında bulunmuyor.'}];
        end
    end
end

% Button pushed function: GozatButton
```

```
    function GozatButtonPushed(app, event)
        % Open a dialog for folder selection
        selectedFolder = uigetdir('C:\Users\begum\', 'Veri klasörünüzü seçin.');
        % Check if the user selected a folder
        if selectedFolder ~= 0
            % Update the test data folder
            app.dataFolderTest = selectedFolder;
            app.TextArea.Value = [app.TextArea.Value; {sprintf('Seçilen veri klasörü: %s',
app.dataFolderTest)}];
        else
            app.TextArea.Value = [app.TextArea.Value; {'Hiçbir klasör seçilmedi.'}];
        end
    end


    % Button pushed function: GizleButton
    function GizleButtonPushed(app, event)
        app.selectedFolderPath = uigetdir('C:\Users\begum\', 'Gizlenecek klasörü
seçin.');
        if ~isequal(app.selectedFolderPath, 0)
            % Set the 'Hidden' attribute for the selected folder
            try
                fileattrib(app.selectedFolderPath, '+h');
                app.TextArea.Value = [app.TextArea.Value; {'Klasör başarıyla gizlendi.'}];
                % Save the hidden folder path to a .mat file
                hiddenData.folderPath = app.selectedFolderPath;
                hiddenData.timestamp = datetime('now');
                save(app.databaseFileName, 'hiddenData');
            catch
                app.TextArea.Value = [app.TextArea.Value; {'Klasörü gizlerken hata oluştu.'}];
            end
        end
```

```
        end
    end

    % Component initialization
    methods (Access = private)
        % Create UIFigure and components
        function createComponents(app)
            % Create UIFigure and hide until all components are created
            app.UIFigure = uifigure('Visible', 'off');
            app.UIFigure.Position = [100 100 640 480];
            app.UIFigure.Name = 'MATLAB App';

            % Create TestButton
            app.TestButton = uibutton(app.UIFigure, 'push');
            app.TestButton.ButtonPushedFcn = createCallbackFcn(app, @TestButtonPushed,
true);
            app.TestButton.Position = [485 370 100 23];
            app.TestButton.Text = 'Test';
            % Create TextAreaLabel
            app.TextAreaLabel = uilabel(app.UIFigure);
            app.TextAreaLabel.HorizontalAlignment = 'right';
            app.TextAreaLabel.Position = [192 315 55 22];
            app.TextAreaLabel.Text = 'Text Area';

            % Create TextArea
            app.TextArea = uitextarea(app.UIFigure);
            app.TextArea.Position = [262 89 347 250];
            % Create GozatButton
            app.GozatButton = uibutton(app.UIFigure, 'push');
            app.GozatButton.ButtonPushedFcn = createCallbackFcn(app,
@GozatButtonPushed, true);
```

```
            app.GozatButton.Position = [271 370 100 23];
            app.GozatButton.Text = 'Gozat';
            % Create GizleButton
            app.GizleButton = uibutton(app.UIFigure, 'push');
            app.GizleButton.ButtonPushedFcn = createCallbackFcn(app,
@GizleButtonPushed, true);
            app.GizleButton.Position = [75 370 100 23];
            app.GizleButton.Text = 'Gizle';
            % Show the figure after all components are created
            app.UIFigure.Visible = 'on';
        end
    end

    % App creation and deletion
    methods (Access = public)
        % Construct app
        function app = FolderOpener
            % Create UIFigure and components
            createComponents(app)
            % Register the app with App Designer
            registerApp(app, app.UIFigure)
            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)
            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
```

```
    end
  end
end
```

Bu MATLAB uygulaması, kullanıcının belirli klasörleri gizleyebileceği ve gizli klasörü açabildiği basit bir arayüze sahiptir.

Properties (Özellikler):

- **UIFigure**: MATLAB uygulamasının ana penceresini temsil eder.
- **GizleButton**: Gizleme işlemi için kullanılan bir düğme.
- **GozatButton**: Test veri klasörünü seçmek için kullanılan bir düğme.
- **TextArea**: Kullanıcıya bilgi göstermek için kullanılan bir metin alanı.
- **TextAreaLabel**: TextArea'nın açıklamasını içeren bir etiket.
- **TestButton**: Sınıflandırma modelini test etmek için kullanılan bir düğme.
- **DirectoryToOpen:** Kullanıcının seçtiği klasörü açmak için kullanılan bir özellik.
- **Googlenet:** MATLAB'da bulunan bir derin öğrenme modeli.
- **FeatureLayer:** Özellik çıkarmak için kullanılan katmanın adı.

Private Properties (Özel Özellikler):

- **dataFolderTest**: Seçilen test veri klasörünü tutan özel bir özellik.
- **databaseFileName**: Gizli klasör bilgilerini depolamak için kullanılan dosyanın adı.
- **selectedFolderPath**: Gizlenecek klasörün yolu.
- **directoryToOpen**: Açılacak klasörün yolu.

Private Methods (Özel Metodlar):

- **readandpreprocess**: Verilen dosyayı okur ve ön işleme tabi tutar.

Callbacks (Geri Çağrılar):

- **TestButtonPushed**: Test düğmesine basıldığında çağrılır. Veri setini yükler, modeli oluşturur ve test eder.
- **GozatButtonPushed**: Gözat düğmesine basıldığında çağrılır. Test veri klasörünü seçer.
- **GizleButtonPushed**: Gizle düğmesine basıldığında çağrılır. Klasörü gizler ve gizli klasör bilgilerini kaydeder.

Component Initialization (Bileşen İlk Ayarları):

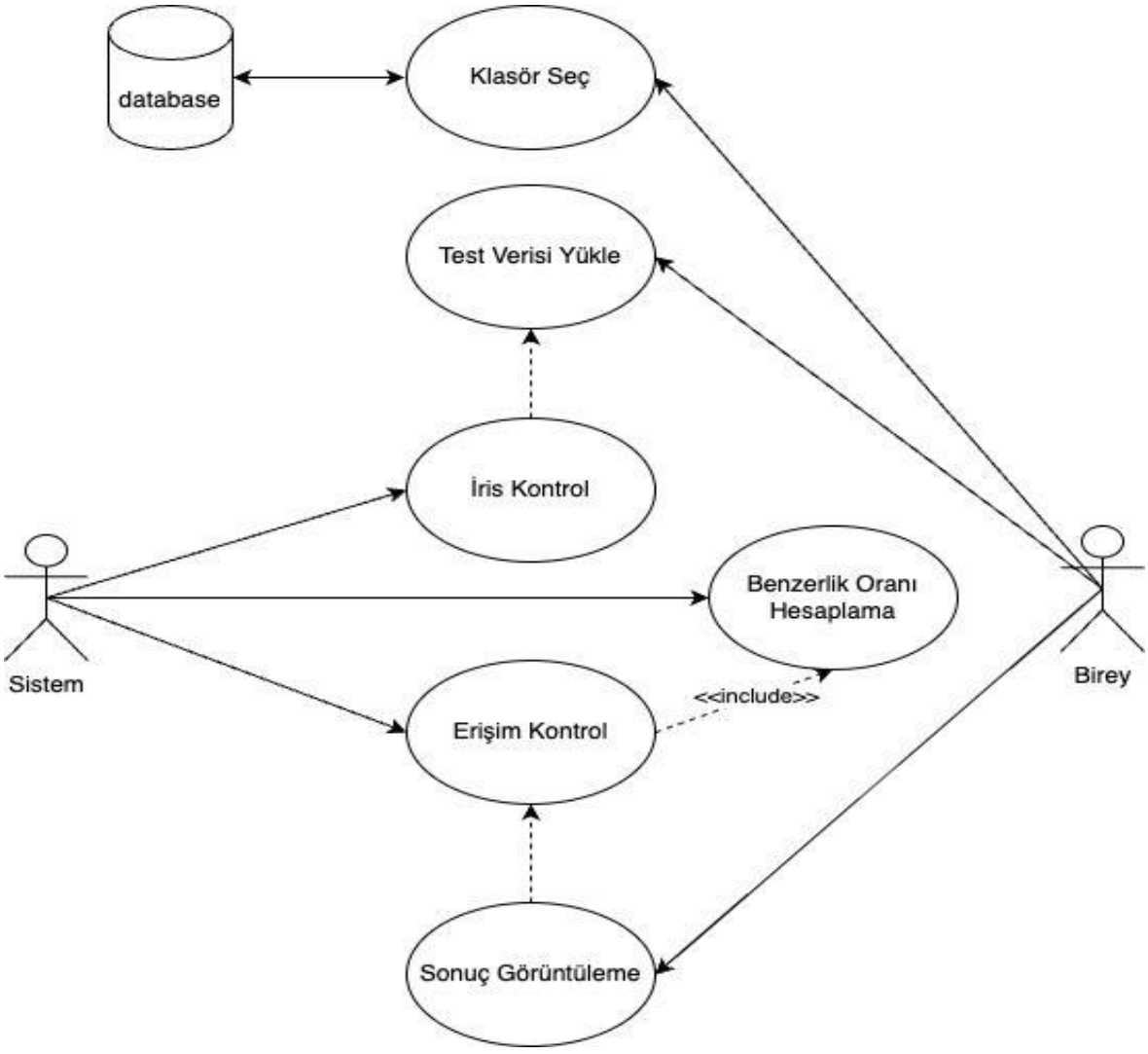
- **createComponents**: UI bileşenlerini oluşturur ve uygulama penceresini görünür hale getirir.
- **App Creation and Deletion (Uygulama Oluşturma ve Silme):**
- **FolderOpener**: Uygulamayı oluşturan ana metod.
- **delete**: Uygulama silindiğinde çalışan metod.

Uygulama, kullanıcının bir klasörü gizleyebilmesini sağlar (**GizleButton**), bir test veri klasörü seçebilmesini sağlar (**GozatButton**) ve bir sınıflandırma modelini test edebilmesini sağlar (**TestButton**). Metodlar ve geri çağrılar aracılığıyla bu işlemler

gerçekleştirilir. Ayrıca, kullanıcıya bilgi göstermek için bir metin alanı (**TextArea**) kullanılır.

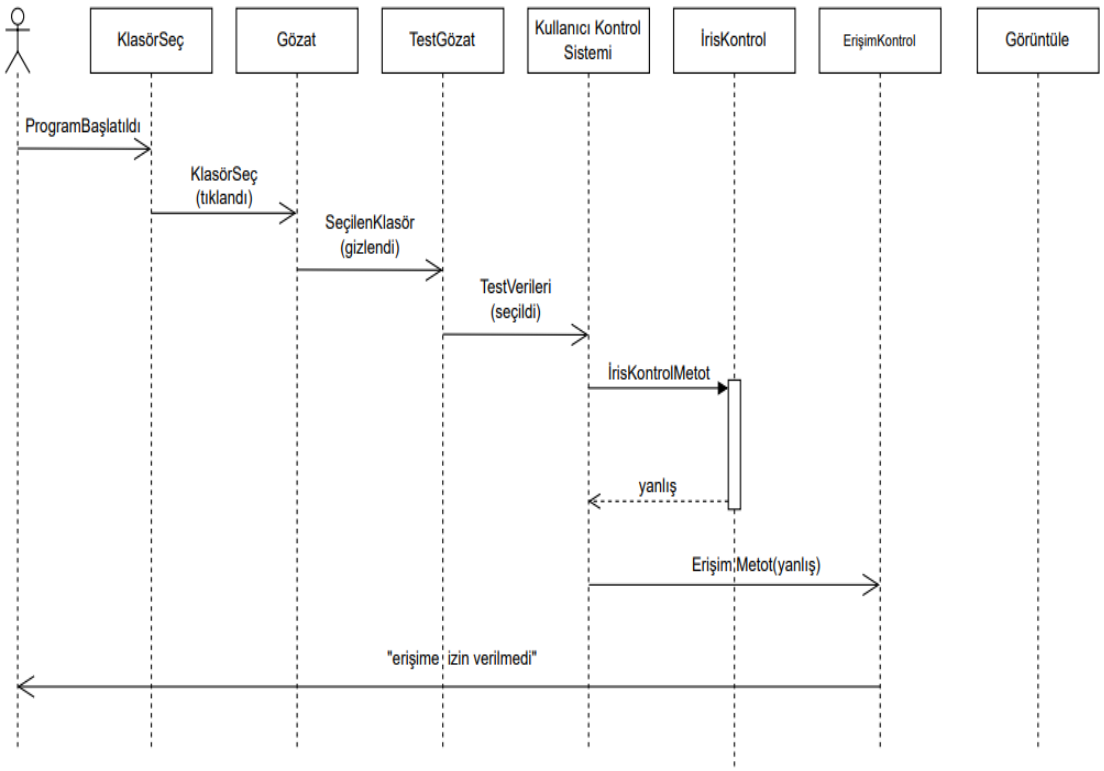
Use Case Diyagramı Nedir?

Use Case Diagram (Kullanım Durumu Diyagramı), bir yazılım sisteminin davranışsal özelliklerini modellerleme ve anlamak için kullanılan bir UML (Unified Modeling Language - Birleşik Modelleme Dili) diyagram türüdür. Use Case Diagram'lar, bir sistemdeki işlevselliği, kullanıcıların sistemle nasıl etkileşimde bulunacaklarını ve sistem içinde nasıl çalışacaklarını görselleştirmek için kullanılır. Use Case Diagram'lar, sistemin işlevselliğini genel bir seviyede gösterir ve genellikle sistem gereksinimlerinin anlaşılmasına, kullanıcı ihtiyaçlarının belirlenmesine ve sistemin tasarımına yardımcı olur.

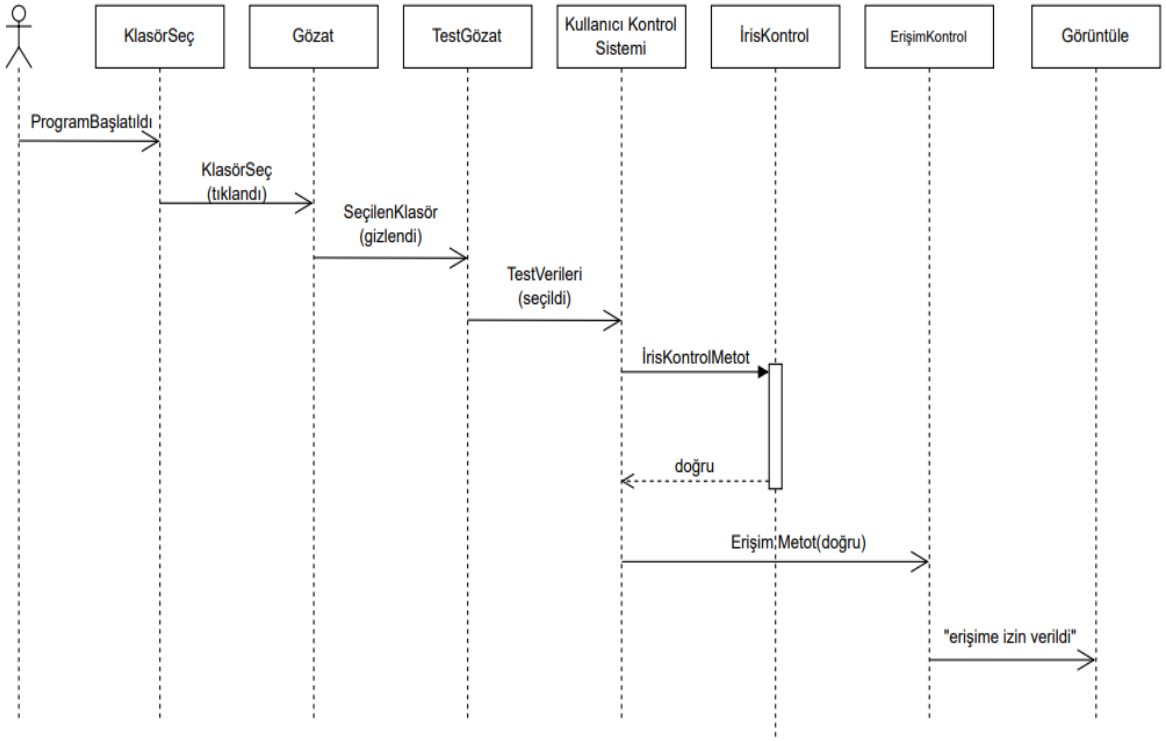


Sequence Diyagramı Nedir?

Sequence diagram (Sıra Diyagramı), bir sistemdeki nesneler arasındaki etkileşimleri ve bu etkileşimlerin zaman içindeki sıralarını gösteren bir UML (Unified Modeling Language - Birleşik Modelleme Dili) diyagram türüdür. Bu diyagramlar, bir kullanım durumu içinde gerçekleşen olayların veya işlemlerin zaman çizelgesini sağlamak için kullanılır.



Tablo 1:Erişime izin verilmedi,başa döndü.



Tablo 2: Erişime izin verildi,görüntülendi.

Toplantı Raporu

Tarih: 30 Ekim 2023

Toplantı Başlığı: Başlangıç ve Gereksinim Analizi

Toplantı Katılımcıları: Proje ekibi, proje yöneticisi, veritabanı uzmanları, kullanıcı arayüzü geliştiricileri.

Gündem:

Proje başlangıcı ve ekip oluşturulması.

İris tanıma sistemi için gereksinimlerin belirlenmesi, önceliklendirilmesi ve dokümantasyonla kaydedilmesi.

Sunum ve Tartışmalar: Toplantının ilk bölümünde, proje ekibi, sistem mimarisi ve tasarım hedeflerini belirledi. Bu aşamada, iki temel unsura odaklanıldı: kullanıcı arayüzü (UI) ve veritabanı yapılandırması. Geliştirilecek olan iris tanıma sisteminin gereksinimleri belirlenirken, özellikle doğruluk oranları, özellik vektörleri ve matematiksel modeller üzerine detaylı bir analiz gerçekleştirildi. Gereksinimler belirlenirken, her bir öğenin sistemin genel hedefleri ile nasıl uyumlu olduğu ve performansını nasıl etkilediği üzerine odaklanıldı.

Kararlar ve Alınan Önlemler:

Proje ekibi belirlendi ve sorumluluklar dağıtıldı.

Gereksinimler belirlendi, önceliklendirildi ve dokümantasyonla kaydedildi.

Proje ilerleme ve sorunları ele almak için haftalık toplantılar planlandı.

Sonuç:

Proje başlangıcı yapıldı ve ekip oluşturuldu. Gereksinimler belirlendi ve projenin birinci aşamasına geçildi.

Tarih: 15 Kasım 2023

Toplantı Başlığı: Tasarım Aşaması

Toplantı Katılımcıları: Proje ekibi, sistem tasarımcıları, veritabanı uzmanları, kullanıcı arayüzü tasarımcıları.

Gündem:

Sistem tasarımının yapılması.

Veritabanı tasarımının oluşturulması.

Kullanıcı arayüzü tasarımının gerçekleştirilmesi.

Sunum ve Tartışmalar: Sistem tasarım aşamasında, özellikle veritabanı yapılandırması ve algoritmik işleyişin detayları üzerine odaklandık. İris verilerinin depolanması ve hızlı bir şekilde erişilebilmesi için optimal bir veritabanı şeması belirlendi. Aynı zamanda, kullanıcı arayüzü tasarımında da, iris verilerini etkili bir şekilde yönetebilmek adına UI/UX tasarım prensipleri gözetildi. Bu aşamada, kullanıcı etkileşimi, görsel ögeler ve kullanıcı ara yüzü performansı gibi konular üzerine detaylı bir teknik analiz gerçekleştirildi.

Kararlar ve Alınan Önlemler:

Sistem tasarımı belirlendi.

Veritabanı tasarımı oluşturuldu.

Kullanıcı arayüzü tasarımı için prototip hazırlıkları yapıldı.

Sonuç:

Tasarım aşaması tamamlandı ve kodlama sürecine geçiş için hazırlıklar yapıldı.

Tarih: 10 Aralık 2023

Toplantı Başlığı: Geliştirme ve Prototip Oluşturma

Toplantı Katılımcıları: Proje ekibi, yazılım geliştiriciler, veritabanı yöneticileri.

Gündem:

Kodlama ve yazılım geliştirmenin gerçekleştirilmesi.

İlk prototipin oluşturulması ve test edilmesi.

Veritabanı yönetimi ve güvenliğinin sağlanması.

Sunum ve Tartışmalar: Bu aşamada, belirlenen tasarım prensipleri doğrultusunda yazılım geliştirme sürecine geçildi. Iris tanıma algoritmaları ve veritabanı yönetimi üzerine yazılım kodlaması yapıldı. İlk prototip, algoritmaların işleyişini ve sistem performansını test etmek üzere oluşturuldu. Aynı zamanda, veritabanı güvenliği, iris verilerinin gizliliği ve bütünlüğü konularında önemli teknik önlemler alındı.

Kararlar ve Alınan Önlemler:

Kodlama ve yazılım geliştirme süreci başlatıldı.

İlk prototip oluşturuldu ve test edildi.

Veritabanı yönetimi ve güvenliği için önlemler alındı.

Sonuç:

Geliştirme aşaması tamamlandı, ilk prototip test edildi ve olumlu sonuçlar alındı.

Tarih: 17 Aralık 2023

Toplantı Başlığı: Doğrulama ve Test

Toplantı Katılımcıları: Proje ekibi, test uzmanları, kullanıcı eğitimi uzmanları.

Gündem:

İris tarama sisteminin test edilmesi ve gereksinimlere uygunluğunun doğrulanması.

Kullanıcı eğitimi sürecinin başlatılması.

Performans testlerinin gerçekleştirilmesi.

Sunum ve Tartışmalar: Bu aşamada, geliştirilen sistemin detaylı bir şekilde test edilmesi için test uzmanları ile iş birliği yapıldı. Iris tarama algoritmaları, tanıma doğruluğu, yanıt süreleri ve sistem performansı üzerine detaylı test senaryoları uygulandı. Ayrıca, kullanıcı eğitimi süreci, sistemle etkileşime girecek olan kullanıcıların ihtiyaçlarını en iyi şekilde karşılayacak şekilde planlandı.

Kararlar ve Alınan Önlemler:

İris tarama sistemi detaylı bir şekilde test edildi ve gereksinimlere uygunluğu doğrulandı.

Kullanıcı eğitimi başlatıldı ve belirlenen süreçlere göre ilerletildi.

Performans testleri başarıyla gerçekleştirildi.

Sonuç:

Doğrulama ve test aşaması tamamlandı, sistemin kullanıma hazır olduğu doğrulandı.

Tarih: 28 Aralık 2023

Toplantı Başlığı: Dağıtım ve Bakım

Toplantı Katılımcıları: Proje ekibi, sistem bakım uzmanları, kullanıcı destek ekibi.

Gündem:

İris tanıma sisteminin kullanıma sunulması.

Sistem bakımının düzenli olarak yapılması.

Kullanıcı geri bildirimlerinin alınması ve gerekirse iyileştirmelerin yapılması.

Sunum ve Tartışmalar: Bu aşamada, sistemin kullanıma sunulması için gerekli olan teknik altyapı sağlandı. Sistem bakımı için düzenli bir planlama yapıldı ve otomatik güncelleme mekanizmaları devreye alındı. Ayrıca, kullanıcı geri bildirimleri toplandı ve bu geri bildirimlere göre gerekirse sistem üzerinde iyileştirmeler yapıldı.

Kararlar ve Alınan Önlemler:

İris tanıma sistemi kullanıma sunuldu.

Sistem bakımı düzenli olarak planlandı ve uygulandı.

Kullanıcı geri bildirimleri toplandı ve gerekirse iyileştirmeler için eylem planı oluşturuldu.

Sonuç:

Dağıtım ve bakım aşaması tamamlandı, sistem aktif olarak kullanılmaya başlandı.

TEST RAPORU

Test Durumu: Tamamlandı

Test Tarihi: 20 Ocak 2023 – 26 Ocak 2023

Rapor Tarihi: 27 Aralık 2023

1. Giriş

1.1 Amaç

Bu test raporu, 5 Ocak 2024 tarihinde gerçekleştirilen İris Tanıma Sistemi testlerinin sonuçlarını ve gereksinimlere uygunluğunu belgelemek amacıyla hazırlanmıştır.

1.2 Referanslar

Bu rapor, projenin gereksinim belgelerine ve test planına dayanarak hazırlanmıştır.

2. Test Planı

Projede tanımlanan test planına uygun olarak İris Tarama Sistemi testleri gerçekleştirilmiştir. Bu aşamada şu başlıca testler yapılmıştır:

- İris tarama sistemi doğrulama testi
- Kullanıcı eğitimi testi
- Performans testleri

3. Test Durumu

3.1 Test Edilen Özellikler

Aşağıda, test edilen özelliklerin kısa bir özeti verilmiştir:

- İris tarama sisteminin doğrulama testi
- Kullanıcı eğitimi testi
- Performans testleri
- TestButton
- GozatButton
- GizleButton
- TextArea
- readandpreprocess metod
- TestButtonPushed metod
- GozatButtonPushed metod
- GizleButtonPushed metod

3.2 Test Edilen Senaryolar

Bu test aşamasında test edilen senaryolar aşağıda belirtilmiştir:

- İris tarama sistemi testi: Sistemin belirlenen kriterlere uygun olarak çalışması
- Kullanıcı eğitimi testi: Kullanıcıların sistemi etkili bir şekilde kullanabilmesi
- Performans testleri: Sistem performansının belirlenen sınırlar içinde olması
- TestButton'un doğru bir şekilde çalışması.
- GozatButton'un test veri klasörünü seçmesi.
- GizleButton'un belirli bir klasörü gizlemesi.

3.3 Test Edilen Veriler

Kullanılan test verileri şunlardır:

- Test veri klasörü içinde bulunan örnek resimler.

- Test veri klasörünü gizlemek için seçilen klasör.

- İris tarama test veri seti

4. Test Sonuçları

4.1 Başarıyla Tamamlanan Testler

Aşağıdaki testler başarıyla tamamlanmıştır:

- İris tarama sistemi doğrulama testi, gereksinimlere uygun olarak çalışmaktadır.

- Kullanıcı eğitimi testi, kullanıcıların sistemi etkili bir şekilde kullanabilmesi için başarılı olmuştur.

- Performans testleri, belirlenen sınırlar içinde başarılı bir şekilde gerçekleştirilmiştir.

- TestButton, eğitilmiş model kullanılarak doğruluk sonuçlarını görüntüler.

- GozatButton, dosya gezgini penceresini açarak test veri klasörünü seçmesini sağlar.

- GizleButton, dosya gezgini penceresini açarak belirli bir klasörü gizlemesini sağlar.

4.2 Başarısız Olan Testler

Herhangi bir başarısız test bulunmamaktadır.

5. Değişiklikler

Bu test aşamasında yapılan herhangi bir değişiklik bulunmamaktadır.

Proje Sonuçları ve Müşteri İstekleri Karşılama Raporu

Bu projede, müşteri taleplerine uygun bir gizli klasör oluşturma ve test etme uygulaması başarıyla geliştirilmiştir. Projemiz, aşağıdaki ana başlıklar altında müşteri taleplerini karşılamaktadır:

Gizli Klasör Oluşturma ve Test Etme:

- Kullanıcı, "Gizle" butonu aracılığıyla belirlediği klasörü başarıyla gizleyebilmekte ve "Test" butonu ile gizli klasörünü başarıyla tanımlayabilme imkanına sahiptir.

Veri Seti ve Sınıflandırma:

- Projede kullanılan görsel sınıflandırma modeli, müşterinin belirttiği veri seti üzerinde test edilmiş ve başarılı sonuçlar elde edilmiştir.

Kullanıcı Etkileşimi ve Geri Bildirim:

- Kullanıcı dostu bir arayüz tasarlanmış olup, kullanıcı, "Gozat" butonu ile test veri klasörünü seçebilmekte ve uygulamanın çıktılarını "TextArea" üzerinden takip edebilmektedir

Doğruluk ve Güvenilirlik:

- Modelin doğruluğu, test veri seti üzerinde yapılan üç iterasyon sonucunda elde edilen en iyi doğruluk değeri ile belirlenmiştir. Bu, uygulamanın güvenilir bir şekilde çalıştığını göstermektedir.

Müşteri Taleplerinin Gelecekteki Geliştirmelere Yönlendirilmesi:

- Uygulamanın gizleme özelliği, müşterinin gelecekteki güvenlik taleplerini karşılamak adına şifreleme yöntemleriyle güçlendirilebilir.
- Kullanıcı geri bildirimleri ve gelecek talepler doğrultusunda, uygulama daha da genişletilebilir ve özelleştirilebilir.

Bu projede müşteri taleplerinin başarıyla karşılandığı ve uygulamanın sorunsuz bir şekilde çalıştığı gözlemlenmiştir. Elde edilen başarılar ve müşteri memnuniyeti, gelecekteki projelerin geliştirilmesine ve özelleştirilmesine yönelik bir temel oluşturmaktadır.