



# ZONGULDAK BÜLENT ECEVİT ÜNİVERSİTESİ

## BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

### BLM497 – BİLGİSAYAR MÜHENDİSLİĞİ PROJE TASARIMI DERSİ

#### ÖN TASARIM BELGESİ (ÖT)

Öğrenciler	Adı Soyadı	Öğrenci No	İmza
	1. Merve Sünbüle	220106109067	<u>M. Sunbule</u>
	2. Şevval Nur Çapur	220106109072	<u>S. Çapur</u>
	3. Rabia Ceylan	220106109023	<u>R. Ceylan</u>
	4. Kamile Burcu İpek	220106109066	<u>K. İpek</u>
Tarih:	27.12.2025		
Proje Danışmanı:	Dr. Öğretim Üyesi Erkan Çetiner		

Projenin Başlığı: Görüntü İşleme ile Zehirli Mantar Tespiti

## 1. Giriş

### 1.1. Amaç

Bu Ön Tasarım Belgesi'nin (ÖT) amacı, Görüntü İşleme ile Zehirli Mantar Tespiti projesi kapsamında geliştirilecek mobil uygulamanın yazılım yapısını, yazılım bileşenlerini, arayüzlerini ve gerçekleme aşamasında kullanılacak veri, algoritma ve işleme süreçlerini ayrıntılı biçimde tanımlamaktır. Bu belge; yazılımin nasıl yapılandırılacağını, hangi bileşenlerden oluşacağını ve bu bileşenlerin birbirleriyle ve kullanıcıyla nasıl etkileşime gireceğini ortaya koyarak gerçekleme aşamasına temel oluşturmayı hedeflemektedir. ÖT'nin hedef kitlesi; projeyi geliştirecek yazılım geliştiriciler, tasarımcılar, test ve doğrulama sürecinde görev alacak kişiler ile proje danışmanıdır. Bu belge, GSÖA'da tanımlanan gereksinimlerin tasarım düzeyinde nasıl karşılandığını açık ve izlenebilir bir biçimde sunarak, geliştirme sürecinde ortak bir teknik referans sağlamayı amaçlamaktadır. Bu kapsamda, Gereksinimler Şartnamesi ve Ön Analiz Belgesi (GSÖA)'nda tanımlanan tüm işlevsel ve işlevsel olmayan gereksinimler, ÖT'de bir veya daha fazla tasarım varlığı ile ilişkilendirilmiş ve izlenebilir hâle getirilmiştir. Her bir gereksinim, yazılım mimarisindeki ilgili modül, bileşen veya arayüz ile eşleştirilerek tasarım kararlarının gereksinimlere dayalı olarak alınması sağlanmıştır. İşlevsel gereksinimler doğrultusunda; kullanıcının uygulama içinden kameralı fotoğraf çekme (FR-1) ve cihaz galerisinden görüntü yükleyebilmesi (FR-2), görüntülerin derin öğrenme modeline uygun olacak şekilde ön işleme adımlarından geçirilmesi (FR-3), TensorFlow Lite tabanlı modelin cihaz üzerinde çalıştırılarak tahmin üretilmesi (FR-4) ve elde edilen sonucun "Zehirli" veya "Yenilebilir" bilgisi ile birlikte güven oranı gösterilerek kullanıcıya sunulması (FR-5) amaçlanmaktadır. Güven oranının belirlenen eşik değerinin altında kaldığı durumlarda kullanıcıya "Emin Değilim" uyarısının verilmesi (FR-6), geçmiş analiz sonuçlarının isteğe bağlı olarak görüntülenebilmesi (FR-7) ve hata veya başarısız işlem durumlarında kullanıcıya açık ve yönlendirici mesajların sunulması (FR-8) da tasarım kapsamında ele alınmaktadır. ÖT ayrıca, kullanıcı arayüzü, donanım arayüzleri, yazılım arayüzleri ve iletişim arayüzlerine ilişkin gereksinimleri kapsamakta; uygulamanın React Native tabanlı, sade ve mobil uyumlu bir arayüze sahip olacak şekilde tasarlanması hedeflemektedir. Gerçekleme aşaması için gerekli olan veri yapıları, model dosyaları, görüntü işleme adımları ve modüller arası veri akışları bu belge kapsamında tanımlanmaktadır.

## **1.2. Kapsam**

Bu Ön Tasarım Belgesi (ÖT), Görüntü İşleme ile Zehirli Mantar Tespiti projesi için hazırlanmış olan gereksinimler Şartnamesi ve Ön Analiz Belgesi (GŞÖA)'nda tanımlanan tüm işlevsel ve işlevsel olmayan gereksinimleri temel alarak, geliştirilecek mobil yazılımın ön tasarımını kapsamaktadır. ÖT, GŞÖA'da belirtilen gereksinimlerin yazılım mimarisi, modüller, arayüzler ve veri akışları düzeyinde nasıl gerçekleştirileceğini açıklamak üzere hazırlanmıştır. Bu belge, GŞÖA'da tanımlanan fotoğraf alma, görüntü işleme, derin öğrenme tabanlı model çalışma ve sonuç gösterimi süreçlerini içeren yazılım sisteminin yapısal ayırtmasını ele almaktadır. Uygulamanın kullanıcı arayüzü, donanım ve yazılım arayüzleri ile iletişim gereksinimleri; GŞÖA'da belirtilen sınırlar doğrultusunda ele alınmış ve yalnızca bu belgede tanımlanan özellikler kapsayacak şekilde sınırlandırılmıştır. ÖT kapsamında, Android ve iOS işletim sistemlerinde çalışacak, React Native tabanlı bir mobil uygulamanın geliştirilmesi hedeflenmektedir. Uygulama, cihaz üzerinde çalışan TensorFlow Lite tabanlı bir derin öğrenme modeli kullanarak mantar görüntülerini analiz edecek; tahmin işlemleri çevrimdışı gerçekleştirilecek ve sistem, sunucu bağımlılığı olmadan çalışacak şekilde tasarlanacaktır. Bu belge, GŞÖA'da tanımlanmayan yeni işlevlerin veya ek sistem bileşenlerinin tasarımını kapsamamaktadır. Ayrıca bu ÖT, gerçekleme (uygulama) aşamasında geliştirilecek yazılım bileşenleri, modüller arası ilişkiler, veri yapıları ve temel işleme adımlarını tanımlamakta; ancak modelin eğitimi, veri setinin oluşturulması veya dağıtım süreçlerine ilişkin ayrıntılı uygulama planlarını kapsam dışı bırakmaktadır. Böylece belge, geliştirilecek yazılımın sınırlarını net bir şekilde ortaya koyarak, tasarım ile gerçekleme aşaması arasında tutarlı ve izlenebilir bir geçiş sağlamayı amaçlamaktadır.

## **1.3. Tanımlar, Kısaltmalar**

### **React Native:**

GŞÖA'da belirtildiği üzere, mobil uygulamanın Android ve iOS platformlarında geliştirilmesi için kullanılan uygulama geliştirme çatısıdır.

### **TensorFlow:**

Derin öğrenme modelinin eğitildiği ve geliştirildiği açık kaynaklı makine öğrenmesi kütüphanesidir.

### **TensorFlow Lite (TFLite):**

Eğitilen derin öğrenme modelinin mobil cihazlarda çalıştırılabilmesi için kullanılan TensorFlow'un optimize edilmiş sürümüdür.

**OpenCV:** Görüntülerin modele uygun hâle getirilmesi için kullanılan açık kaynaklı görüntü işleme kütüphanesidir.

**NumPy:** Model geliştirme ve görüntü işleme süreçlerinde kullanılan sayısal hesaplama kütüphanesidir.

**Expo:** GŞÖA'da belirtildiği üzere, React Native tabanlı uygulamanın geliştirilmesi sürecinde kullanılan yardımcı araç setidir.

**Visual Studio Code:** Uygulamanın geliştirilmesi sırasında kullanılan kod geliştirme ortamıdır.

**Android:** Uygulamanın çalışacağı mobil işletim sistemlerinden biridir.

**iOS:** Uygulamanın çalışacağı mobil işletim sistemlerinden biridir.

## 2. Kaynakça

### Gereksinimler Şartnamesi ve Ön Analiz Belgesi (GŞÖA)

BLM497 – Bilgisayar Mühendisliği Proje Tasarımı Dersi,

Tarih: 30.10.2025,

Yayımlayan Kuruluş: BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ,

Erişim Ortamı: Ders kapsamında öğrencilere sağlanan PDF dokümanı.

### TensorFlow Lite Documentation

Başlık: *TensorFlow Lite Guide*,

Yayımlayan Kuruluş: Google LLC,

Erişim Ortamı: Resmî TensorFlow Lite çevrim içi dokümantasyonu.

### TensorFlow Documentation

Başlık: *TensorFlow Machine Learning Platform*,

Yayımlayan Kuruluş: Google LLC,

Erişim Ortamı: Resmî TensorFlow çevrim içi dokümantasyonu.

### OpenCV Documentation

Başlık: *OpenCV Library Documentation*,

Yayımlayan Kuruluş: OpenCV Foundation,

Erişim Ortamı: OpenCV resmî çevrim içi dokümantasyonu.

### React Native Documentation

Başlık: *React Native Documentation*,

Yayımlayan Kuruluş: Meta Platforms Inc.,

Erişim Ortamı: React Native resmî çevrim içi dokümantasyonu.

## **Expo Documentation**

Başlık: *Expo Application Services Documentation*,

Yayımlayan Kuruluş: Expo Dev Team,

Erişim Ortamı: Expo resmî çevrim içi dokümantasyonu.

## **Kaggle – Mushroom Image Datasets**

Başlık: *Mushroom Classification Image Datasets*,

Yayımlayan Kuruluş: Kaggle Inc.,

Erişim Ortamı: Kaggle veri paylaşım platformu (çevrim içi).

## **ChatGPT**

Başlık: OpenAI ChatGPT – GPT Tabanlı Dil Modeli ve Dokümantasyon Yardımı

Yayımlayan Kuruluş: OpenAI

Erişim Ortamı: Çevrim içi yapay zeka destekli metin üretim ve danışmanlık platformu

## **3. Tasarım Varlıklarının Özellikleri**

### **3.1. Kimlik Belirleme**

ImageAcquisitionModule: Kullanıcının kamera veya galeri aracılığıyla görüntü aktarmasını sağlayan tasarım varlığıdır.

ImagePreprocessingModule: Alınan görüntü üzerinde ön işleme işlemlerini gerçekleştiren tasarım varlığıdır.

ModelInferenceModule: TensorFlow Lite tabanlı modeli çalıştırarak tahmin üreten tasarım varlığıdır.

ResultPresentationModule: Tahmin sonuçlarını kullanıcıya sunan ve gerekli uyarıları gösteren tasarım varlığıdır.

### **3.2. Tür**

ImageAcquisitionModule: Kullanıcı etkileşimi ile tetiklenen, mobil cihazın kamera ve galeri servislerine erişim sağlayan, görüntü alma işlevlerini yerine getiren uygulama katmanı yazılım modülü.

ImagePreprocessingModule: Ham görüntü verisi üzerinde yeniden boyutlandırma ve normalizasyon gibi dönüşümler gerçekleştiren, hesaplama odaklı yardımcı yazılım modülü.

ModelInferenceModule: TensorFlow Lite tabanlı derin öğrenme modelini çalıştırarak tahmin üreten, uygulamanın çekirdek işlevini yerine getiren model yürütme yazılım modülü.

**ResultPresentationModule:** Modelden elde edilen tahmin sonuçlarını kullanıcıya görsel ve metinsel biçimde sunan, uyarı ve bilgilendirme işlevleri içeren kullanıcı arayüzü odaklı yazılım modülü.

### **3.3. Amaç**

**ImageAcquisitionModule:** Bu tasarım varlığının oluşturulma amacı, GŞÖA'da tanımlanan FR-1 (Fotoğraf Çekme) ve FR-2 (Görüntü Yükleme) işlevsel gereksinimlerini karşılamaktır.

Bu varlık, kullanıcının mobil cihaz kamerası veya galeri aracılığıyla mantar görüntüsünü sisteme güvenilir ve kontrollü bir şekilde aktarmasını sağlamak amacıyla tasarlanmıştır.

Bu sayede sistemin görüntüye dayalı tahmin sürecinin başlatılması mümkün hâle gelmektedir.

**ImagePreprocessingModule:** Bu tasarım varlığının oluşturulma amacı, GŞÖA'da belirtilen FR-3 (Görüntü İşleme) işlevsel gereksinimini karşılamaktır.

Bu varlık, alınan ham görüntüyü derin öğrenme modelinin giriş formatına uygun hâle getirerek modelin doğru ve tutarlı biçimde çalışabilmesi için gerekli ön koşulları sağlamaktadır.

**ModelInferenceModule:** Bu tasarım varlığının oluşturulma amacı, GŞÖA'da tanımlanan FR-4 (Model Çalıştırma) işlevsel gereksinimini karşılamaktır.

Bu varlık, TensorFlow Lite formatındaki derin öğrenme modelini çalıştırarak mantarın zehirli veya yenilebilir olduğuna dair tahmin üretmek ve sistemin temel karar mekanizmasını gerçekleştirmek üzere tasarlanmıştır.

**ResultPresentationModule:** Bu tasarım varlığının oluşturulma amacı, GŞÖA'da tanımlanan FR-5 (Sonuç Gösterimi), FR-6 (Eşik ve Uyarılar), FR-7 (Sonuç Kaydı) ve FR-8 (Hata Yönetimi) işlevsel gereksinimlerini karşılamaktır.

Bu varlık, model tarafından üretilen tahmin sonuçlarını kullanıcıya anlaşılır bir biçimde sunmak, belirsizlik durumlarında kullanıcıyı uyarmak ve sonuçların istege bağlı olarak saklanmasılığını sağlamak amacıyla tasarlanmıştır.

### **3.4. İşlev (fonksiyon)**

**ImageAcquisitionModule:** Bu varlık, kullanıcıdan gelen etkileşim doğrultusunda kamera veya galeri üzerinden bir görüntü girdisi alır.

Alınan bu ham görüntü verisi üzerinde herhangi bir içeriksel dönüşüm gerçekleştirilmeden, sistemin bir sonraki aşaması olan görüntü ön işleme modülüne iletılır.

**ImagePreprocessingModule:** Bu varlık, ham görüntü verisini girdi olarak alır ve derin öğrenme modelinin kabul ettiği giriş formatına dönüştürür.

Bu dönüşüm; görüntünün yeniden boyutlandırılması, renk uzayının düzenlenmesi ve normalizasyon işlemlerini kapsar. Çıktı olarak, modele uygun hâle getirilmiş işlenmiş görüntü verisi üretir.

**ModelInferenceModule:** Bu varlık, işlenmiş görüntü verisini girdi olarak alır ve TensorFlow Lite tabanlı derin öğrenme modelini çalıştırır.

Model çıktıları üzerinden sınıflandırma işlemi gerçekleştirilir ve mantarın zehirli veya yenilebilir olduğuna dair tahmin ile buna ait güven oranı üretılır.

**ResultPresentationModule:** Bu varlık, model tahmin sonuçlarını girdi olarak alır ve sonuçları kullanıcıya görsel ve metinsel biçimde sunar.

Tahmin sonucuna bağlı olarak eşik kontrollerini gerçekleştirir, gerekli uyarı mesajlarını üretir ve isteğe bağlı olarak sonuçların kayıt altına alınmasını sağlar.

### **3.5. Altparçalar**

**ImageAcquisitionModule:**

Bu tasarım varlığı, görüntü alma işlevlerini gerçekleştirebilmek için aşağıdaki altparçalardan oluşmaktadır:

- Kamera Erişim Altparçası
- Mobil cihaz kamerası üzerinden fotoğraf çekme işlemini gerçekleştirir.
- Galeri Erişim Altparçası
- Cihaz depolama alanından mevcut bir görüntünün seçilmesini sağlar.

Bu altparçalar, ImageAcquisitionModule varlığının çocuk bileşenleri olup, görüntü edinme gereksinimlerinin izlenebilirliğini sağlar.

**ImagePreprocessingModule.**

Bu tasarım varlığı, görüntü ön işleme sürecini aşağıdaki altparçalara ayırarak gerçekleştirir:

- Yeniden Boyutlandırma Altparçası
- Görüntüyü modelin kabul ettiği boyuta dönüştürür.
- Normalizasyon Altparçası
- Piksel değerlerini belirlenen aralığa ölçekler.

Bu altparçalar, görüntü işleme adımlarının ayrıştırılmasını ve gereksinimlerin alt düzeyde izlenmesini sağlar.

**ModelInferenceModule:**

Bu tasarım varlığı, model tahmin sürecini aşağıdaki altparçalar üzerinden gerçekleştirir:

- Model Yükleme Altparçası
- TensorFlow Lite model dosyasını belleğe yükler.
- Tahmin Çalıştırma Altparçası
- Yüklenen modeli kullanarak tahmin işlemini gerçekleştirir.

Bu yapı, modelin yönetimi ile tahmin sürecinin birbirinden ayrılmasını sağlar.

ResultPresentationModule.

Bu tasarım varlığı, sonuçların kullanıcıya sunulmasını sağlayan aşağıdaki altparçalardan oluşmaktadır:

- Sonuç Gösterim Altparçası
- Tahmin sonucunu ve güven oranını kullanıcı arayüzünde görüntüler.
- Uyarı ve Eşik Kontrol Altparçası
- Belirsizlik durumlarını değerlendirir ve kullanıcıyı bilgilendirir.
- Sonuç Kayıt Altparçası
- Kullanıcının isteğine bağlı olarak analiz sonuçlarını saklar.

Bu altparçalar, sonuç sunumu ve yönetimi işlevlerinin modüler bir yapıda izlenmesini sağlar.

### 3.6. Bağımlılıklar

ImageAcquisitionModule:

Bu tasarım varlığı, kullanıcı arayüzü üzerinden alınan etkileşimle ilk değer atama süreci başlatılarak çalıştırılır.

Kamera veya galeri erişimi için gerekli izinlerin durumu, modül başlatılırken ilk değer olarak değerlendirilir.

ImageAcquisitionModule, çalışması sonucunda ham görüntü verisini yaratır ve bu veriyi ImagePreprocessingModule varlığına aktarır.

Bu nedenle, ImageAcquisitionModule sisteme ilk çalışan modül olup, diğer modüllerin çalışabilmesi için koşma sırası açısından ön koşul oluşturur.

ImagePreprocessingModule:

Bu tasarım varlığı, ImageAcquisitionModule tarafından yaratılan ham görüntü verisini kullanır.

Modül başlatıldığında, modelin kabul ettiği giriş boyutu ve normalizasyon parametreleri ilk değerler olarak atanır.

ImagePreprocessingModule, işleme sonucunda işlenmiş görüntü verisini oluşturur ve bu veriyi ModelInferenceModule varlığına iletir.

Bu modül, görüntü alma modülünün tamamlanmasına bağlıdır ve ondan **sonra** çalıştırılır.

ModelInferenceModule:

Bu tasarım varlığı, ImagePreprocessingModule tarafından üretilen işlenmiş görüntü verisini girdi olarak kullanır.

Modül başlatılırken TensorFlow Lite model dosyası belleğe yüklenir ve gerekli çalışma parametreleri ilk değer atama sürecinde belirlenir.

ModelInferenceModule, tahmin işlemi sonucunda sınıflandırma sonucu ve güven oranı verisini yaratır ve bu veriyi ResultPresentationModule varlığına aktarır.

Bu varlık, ön işleme modülünün tamamlanmasına bağlı olarak çalışır.

ResultPresentationModule:

Bu tasarım varlığı, ModelInferenceModule tarafından üretilen tahmin sonuçlarını kullanır.

Modül başlatıldığında, eşik değerleri ve uyarı koşulları ilk değerler olarak atanır.

ResultPresentationModule, tahmin sonuçlarını kullanıcıya sunar ve kullanıcının tercihine bağlı olarak sonuç verilerini cihaz depolama alanına kaydedebilir.

Bu süreçte, veri depolama ve daha sonra gerektiğinde veri kullanımına ilişkin bağımlılıklar oluşur.

### 3.7. Arayüz

ImageAcquisitionModule

Bu tasarım varlığı, kullanıcı arayüzü üzerinden diğer varlıklarla etkileşime girer.

Etkileşim, kullanıcı tarafından tetiklenen arayüz bileşenleri aracılığıyla yönetilir.

Girdiler

- Kullanıcı komutu (Fotoğraf çek / Galeriden seç)
  - Tür: Kullanıcı olayı
  - Biçim: UI olay çağrıları

Girdi Erimi

- Kamera ve/veya depolama erişim izinlerinin verilmiş olması

Çıktılar

Ham görüntü verisi

- Tür: Görüntü verisi
- Biçim: RGB formatında dijital görüntü

Çıktı Hata Kodları

**E-IA-01:** Kamera erişim izni reddedildi.

**E-IA-02:** Galeri erişimi reddedildi.

**E-IA-03:** Görüntü alma işlemi kullanıcı tarafından iptal edildi.

## ImagePreprocessingModule

Bu tasarım varlığı, ImageAcquisitionModule tarafından sağlanan görüntü verisi ile etkileşime girer.

Etkileşim, fonksiyon çağrıları yoluyla yönetilir.

### Girdiler

Ham görüntü verisi

- Tür: Görüntü verisi
- Biçim: RGB formatında dijital görüntü

### Girdi Erimi

Okunabilir ve bozulmamış görüntü verisi

### Çıktılar

İşlenmiş görüntü verisi

- Tür: Sayısal görüntü verisi
- Biçim: Normalize edilmiş tensör

### Çıktı Hata Kodları

**E-IP-01:** Geçersiz görüntü formatı..

**E-IP-02:** Görüntü ön işleme başarısız.

**E-IP-03:** Bellek yetersizliği.

## ModelInferenceModule

Bu tasarım varlığı, **ImagePreprocessingModule** ile veri alışverişesi yapar.

Etkileşim, model çalışma çağrıları ile yönetilir.

### Girdiler

İşlenmiş görüntü verisi

- Tür: Sayısal veri
- Biçim: Tensör

### Girdi Erimi

Modelin kabul ettiği boyut ve değer aralığına uygunluk

#### Çıktılar

##### Tahmin sonucu

- Tür: Sınıflandırma etiketi (Zehirli / Yenilebilir / Emin Değilim)
- Biçim: Metinsel değer

##### Güven oranı

- Tür: Sayısal değer
- Biçim: Yüzdelik oran

#### Çıktı Hata Kodları

**E-MI-01:** Model dosyası yüklenemedi.

**E-MI-02:** Model çalışma hatası..

**E-MI-03:** Geçersiz girdi boyutu.

#### ResultPresentationModule

Bu tasarım varlığı, ModelInferenceModule tarafından üretilen çıktılarla etkileşime girer.

Etkileşim, kullanıcı arayüzü üzerinden sonuçların görselleştirilmesiyle yönetilir.

#### Girdiler

- Tahmin sonucu
- Güven oranı

#### Girdi Erimi

- Geçerli tahmin etiketi ve 0–85 aralığında güven oranı

#### Çıktılar

- Sonuç ekranı
  - Tür: Görsel ve metinsel çıktı
  - Biçim: Mobil uygulama arayüzü

#### Çıktı Hata Kodları

**E-RP-01:** Sonuç ekranı oluşturulamadı

**E-RP-02:** Sonuç kaydı başarısız

**E-RP-03:** Görselleştirme hatası

### **3.8. Kaynaklar**

ImageAcquisitionModule

Bu tasarım varlığının işlevini yerine getirebilmesi için aşağıdaki tasarım dışı kaynaklara ihtiyaç duyulmaktadır:

#### **Fiziksel Kaynaklar:**

- Mobil cihaz kamerası
- Cihaz depolama alanı

#### **Yazılım Hizmetleri:**

- Mobil işletim sistemi kamera servisleri
- Dosya sistemi erişim servisleri

#### **İşleme Kaynakları:**

- Kamera erişimi sırasında CPU çevrimleri
- Görüntü alımı için geçici bellek kullanımı

ImagePreprocessingModule

Bu tasarım varlığı, görüntü işleme işlemlerini gerçekleştirebilmek için aşağıdaki kaynaklara ihtiyaç duyar:

#### **Fiziksel Kaynaklar:**

- Cihaz belleği (RAM)

#### **Yazılım Hizmetleri:**

- OpenCV görüntü işleme kütüphanesi

#### **İşleme Kaynakları:**

- Görüntü yeniden boyutlandırma ve normalizasyon işlemleri için CPU çevrimleri
- Ara tensörler için bellek ayırma

ModelInferenceModule

Bu tasarım varlığının çalışabilmesi için aşağıdaki kaynaklar gereklidir:

### Fiziksel Kaynaklar:

- Cihaz belleği
- Yerel depolama (model dosyası için)

### Yazılım Hizmetleri:

- TensorFlow Lite çalışma ortamı
- Donanım hızlandırma servisleri (NNAPI / Metal, mevcutsa)

### İşleme Kaynakları:

- Model çalışma sırasında CPU veya donanım hızlandırıcı çevrimleri
- Tahmin süreci için bellek ayırma

ResultPresentationModule

Bu tasarım varlığı, sonuçların kullanıcıya sunulabilmesi için aşağıdaki kaynakları kullanır:

### Fiziksel Kaynaklar:

- Mobil cihaz ekranı
- Cihaz depolama alanı (sonuç kayıtları için)

### Yazılım Hizmetleri:

- React Native kullanıcı arayüzü bileşenleri
- Mobil işletim sistemi grafik arayüz servisleri

### İşleme Kaynakları:

- Arayüz güncellemeleri için CPU çevrimleri
- Sonuç verilerinin yönetimi için bellek kullanımı

## 3.9. İşleme

ImageAcquisitionModule

Bu tasarım varlığı, kullanıcıdan gelen etkileşim olayını algılayarak çalışmaya başlar.

İşleme süreci aşağıdaki adımlardan oluşur:

1. Kullanıcının “Fotoğraf Çek” veya “Galeriden Seç” seçeneğini tetiklemesi,
2. Gerekli kamera veya depolama izinlerinin kontrol edilmesi,
3. İzinler mevcutsa ilgili servislerin çağrılması,
4. Görüntünün alınması ve ham görüntü verisinin oluşturulması,
5. Ham görüntü verisinin ImagePreprocessingModule varlığına iletilmesi.

İşleme süreci, görüntünün başarıyla alınması veya kullanıcının işlemi iptal etmesi durumunda sonlanır.

#### ImagePreprocessingModule

Bu tasarım varlığı, ham görüntü verisini girdi olarak alarak ön işleme algoritmalarını uygular.

İşleme süreci aşağıdaki adımları içerir:

1. Girdi görüntüsünün geçerlilik kontrolünün yapılması,
2. Görüntünün modelin kabul ettiği boyuta yeniden ölçeklendirilmesi,
3. Renk formatının düzenlenmesi,
4. Piksel değerlerinin normalizasyonunun gerçekleştirilmesi,
5. İşlenmiş görüntü verisinin oluşturulması ve ModelInferenceModule varlığına aktarılması.

İşleme süreci, görüntünün başarıyla işlenmesi veya geçersiz girdi tespit edilmesi durumunda sonlandırılır.

#### ModelInferenceModule

Bu tasarım varlığı, işlenmiş görüntü verisini kullanarak derin öğrenme modelini çalıştırır.

İşleme süreci aşağıdaki adımlardan oluşur:

1. TensorFlow Lite modelinin belleğe yüklenmesi,
2. Girdi verisinin model giriş katmanına beslenmesi,
3. Modelin ileri besleme (inference) işlemini gerçekleştirmesi,
4. Çıktı katmanından sınıflandırma sonuçlarının elde edilmesi,
5. Güven oranının hesaplanması.

İşleme süreci, tahmin sonucunun üretilmesiyle tamamlanır veya model çalışma hatası durumunda sonlandırılır.

#### ResultPresentationModule

Bu tasarım varlığı, model tahmin sonuçlarını kullanıcıya sunmak için aşağıdaki işlem adımlarını uygular:

1. Tahmin sonucu ve güven oranının alınması,
2. Önceden belirlenen eşik değerleriyle karşılaştırma yapılması,
3. Belirsizlik durumlarının değerlendirilmesi,
4. Sonucun görsel ve metinsel olarak kullanıcı arayüzünde gösterilmesi,
5. Kullanıcının tercihine bağlı olarak sonuçların kayıt altına alınması.

İşleme süreci, sonuç ekranının kullanıcıya başarıyla sunulmasıyla sonlanır.

### 3.10. Veri

#### ImageAcquisitionModule

Bu tasarım varlığı tarafından kullanılan veri, kullanıcıdan alınan ham görüntü verisidir.

- **Gösterim Yöntemi:** Dijital görüntü dosyası.
- **İlk Değer:** Başlangıçta boş (null); kullanıcı tarafından görüntü alındığında oluşturulur.
- **Kullanım:** Görüntü ön işleme modülüne girdi olarak aktarılır.
- **Veri Biçimi:** RGB renk uzayında, sıkıştırılmış veya sıkıştırılmamış görüntü formatı. (ör. JPEG/PNG)
- **Kabul Edilebilir Değerler:** Okunabilir, bozulmamış ve geçerli görüntü dosyaları.

### ImagePreprocessingModule

Bu tasarım varlığı, ham görüntü verisini işleyerek işlenmiş görüntü verisi üretir.

- **Gösterim Yöntemi:** Sayısal veri dizisi.
- **İlk Değer:** Ham görüntü alındığında oluşturulur.
- **Kullanım:** Model tahmin modülüne girdi olarak kullanılır.
- **Veri Biçimi:** Normalize edilmiş tensör. (ör.  $224 \times 224 \times 3$ )
- **Kabul Edilebilir Değerler:** Modelin giriş katmanı tarafından kabul edilen boyut ve değer aralıkları.

### ModelInferenceModule

Bu tasarım varlığı tarafından kullanılan veri, model tahmin çıktılarıdır.

- **Gösterim Yöntemi:**
- Mantıksal ve sayısal veri kombinasyonu
- **İlk Değer:**
- Model çalıştırılmadan önce tanımsızdır; tahmin sonucunda oluşturulur.
- **Kullanım:**
- Sonuç gösterim modülüne aktarılır.
- **Veri Biçimi:**
  - Sınıflandırma etiketi (metinsel)
  - Güven oranı (sayısal, yüzdelik)
- **Kabul Edilebilir Değerler:**
  - Etiket: “Zehirli”, “Yenilebilir” veya “Emin Değilim”
  - Güven oranı: %0 – %85 aralığında

### ResultPresentationModule

Bu tasarım varlığı, tahmin sonuçlarını kullanarak sonuç verisi üretir ve yönetir.

- **Gösterim Yöntemi:** Görsel ve metinsel kullanıcı arayüzü çıktıları.
- **İlk Değer:** Tahmin sonucu alındığında oluşturulur.
- **Kullanım:** Kullanıcıya bilgi sunmak ve istege bağlı olarak geçmiş sonuçları saklamak için kullanılır.
- **Veri Biçimi:** Ekran çıktıları ve yerel kayıt verileri.

- **Kabul Edilebilir Değerler:** Geçerli tahmin sonucu ve güven oranı içeren kayıtlar.

## 4. Ayristirma Aclarlamasi

### 4.1. Genel Yapı

Bu Ön Tasarım belgesinde tanımlanan yazılım sistemi, GŞÖA'da belirtilen işlevsel gereksinimler doğrultusunda modüler bir yapı esas alınarak ayristirılmıştır.

Sistem, her biri belirli bir sorumluluğu yerine getiren ve aralarında açık veri akışı bulunan dört ana tasarım varlığından oluşmaktadır.

Ayristirma, sistemin görüntüye dayalı mantar tanıma sürecini uçtan uca destekleyecek şekilde, ardışık çalışan modüller temelinde gerçekleştirilmiştir.

ImageAcquisitionModule

**Kimlik:** ImageAcquisitionModule

**Tür:** Kullanıcı etkileşimi ile tetiklenen yazılım modülü

Bu varlık, kullanıcının kamera veya galeri aracılığıyla mantar görüntüsünü sisteme aktarmasını sağlar.

GŞÖA'da tanımlanan FR-1 (Fotoğraf Çekme) ve FR-2 (Görüntü Yükleme) gereksinimlerini karşılar.

Bu modül, sistemde çalışan ilk bileşen olup, görüntü alma sürecini başlatır.

Bu varlığın ayrıntılı tanımı için Bölüm 3.1 – 3.10 (ImageAcquisitionModule) başvurulmalıdır.

ImagePreprocessingModule

**Kimlik:** ImagePreprocessingModule

**Tür:** Hesaplama odaklı yardımcı yazılım modülü

Bu varlık, alınan ham görüntüyü derin öğrenme modeline uygun hâle getirmek için gerekli ön işleme adımlarını uygular.

GŞÖA'da belirtilen FR-3 (Görüntü İşleme) gereksinimini karşılar.

Bu modül, görüntü alma işleminin ardından çalışır ve model tahmini için ön koşul oluşturur.

Bu varlığın ayrıntılı tanımı için Bölüm 3.1 – 3.10 (ImagePreprocessingModule) başvurulmalıdır.

#### ModelInferenceModule

**Kimlik:** ModelInferenceModule

**Tür:** Model yürütme yazılım modülü

Bu varlık, TensorFlow Lite tabanlı derin öğrenme modelini çalıştırarak mantarın zehirli veya yenilebilir olduğuna dair tahmin üretir.

GŞÖA'da tanımlanan FR-4 (Model Çalıştırma) gereksinimini karşılar.

Sistem mimarisinde çekirdek işlevi yerine getiren modüldür.

Bu varlığın ayrıntılı tanımı için Bölüm 3.1 – 3.10 (ModelInferenceModule) başvurulmalıdır.

#### ResultPresentationModule

**Kimlik:** ResultPresentationModule

**Tür:** Kullanıcı arayüzü odaklı yazılım modülü

Bu varlık, model tarafından üretilen tahmin sonuçlarını kullanıcıya sunar, belirsizlik durumlarını yönetir ve isteğe bağlı olarak sonuç kayıtlarını tutar.

GŞÖA'da tanımlanan FR-5, FR-6, FR-7 ve FR-8 gereksinimlerini karşılar.

Sistemin kullanıcıyla doğrudan etkileşimde bulunan son aşamasıdır.

Bu varlığın ayrıntılı tanımı için Bölüm 3.1 – 3.10 (ResultPresentationModule) başvurulmalıdır.

### 4.1.1. Modül Ayrıştırma

Bu alt bölümde, yazılım sistemi Bölüm 3'te tanımlanan tasarım varlıkları temel alınarak yazılım modülleri seviyesinde ayrıştırılmıştır. Ayrıştırma, GŞÖA'da belirtilen işlevsel gereksinimlerin her birinin en az bir yazılım modülü tarafından karşılanması sağlanacak biçimde yapılmıştır.

Yazılım sistemi, aşağıda açıklanan dört ana yazılım modülünden oluşmaktadır. Her modül, kendi işlevini yerine getirmek için mantıksal olarak bağımsız alt parçalara ayrılmıştır. Alt parçalar, modülün sorumluluklarını daha küçük, yönetilebilir ve test edilebilir birimlere böler; böylece bakım, genişletme ve hata ayıklama süreçleri kolaylaşır.

## ImageAcquisitionModule

- **Kimlik:** ImageAcquisitionModule
- **Tür:** Kullanıcı etkileşimli yazılım modülü
- **Amaç:** Kullanıcının kamera veya galeri yoluyla mantar görüntüsünü sisteme aktarmasını sağlamak.
- **İşlev:** Kamera veya depolama biriminden görüntü alarak sistem içinde kullanılabilir dijital görüntü verisi üretmek.
- **Altparçalar ve Ayırıştırma Gerekçesi:**
  - **Kamera Erişim Bileşeni:** Kameradan görüntü almayı sağlar; bu işlem, depolama erişiminden bağımsızdır.
  - **Galeri Erişim Bileşeni:** Kullanıcının mevcut fotoğrafları seçmesini sağlar; kamera erişiminden bağımsızdır.
  - **İzin Yönetimi Bileşeni:** Kamera ve galeri erişim izinlerini yönetir; diğer alt parçaların işlevsellliğini etkilemeden çalışabilir.
- **Karşıladığı Gereksinimler:**
  - FR-1: Fotoğraf Çekme
  - FR-2: Görüntü Yükleme

Alt parçalar, farklı kaynaklardan veri almayı ve kullanıcı izinlerini ayrı ayrı yönetmeyi kolaylaştırmak amacıyla bağımsız şekilde tasarlanmıştır.

## ImagePreprocessingModule

- **Kimlik:** ImagePreprocessingModule
- **Tür:** Hesaplama odaklı yazılım modülü
- **Amaç:** Alınan ham görüntüyü, makine öğrenmesi modelinin giriş gereksinimlerine uygun hâle getirmek.
- **İşlev:** Görüntüyü yeniden boyutlandırma, renk uzayı dönüştürme ve normalizasyon işlemlerinden geçirerek işlenmiş görüntü üretmek.
- **Altparçalar ve Ayırıştırma Gerekçesi:**
  - **Yeniden Boyutlandırma Bileşeni:** Görüntüyü modelin giriş boyutlarına göre ayarlar; diğer ön işleme işlemlerinden bağımsızdır.
  - **Normalizasyon Bileşeni:** Piksel değerlerini modelin gereksinimlerine göre dönüştürür; yeniden boyutlandırmadan bağımsız çalışabilir.
  - **Görüntü Doğrulama Bileşeni:** İşlenmiş görüntünün geçerliliğini kontrol eder; hata yönetimini kolaylaştırır ve diğer bileşenlerden bağımsızdır.
- **Karşıladığı Gereksinimler:**
  - FR-3: Görüntü İşleme

Alt parçalar, farklı ön işleme adımlarının bağımsız test edilebilmesi ve yeniden kullanılabilir olması için ayırtırılmıştır.

## ModelInferenceModule

- **Kimlik:** ModelInferenceModule
- **Tür:** Model yürütme yazılım modülü
- **Amaç:** İşlenmiş görüntü üzerinde derin öğrenme modelini çalıştırarak sınıflandırma sonucu üretmek.

- **İşlev:** TensorFlow Lite modelini kullanarak “Zehirli / Yenilebilir” tahmini ve güven oranını hesaplamak.
- **Altparçalar ve Ayristirma Gerekcesi:**
  - **Model Yükleme Bileşeni:** Modelin belleğe yüklenmesini sağlar; tahmin motorundan bağımsızdır.
  - **Tahmin Motoru:** Modeli çalıştırarak tahmin üretir; yükleme ve eşik değerlendirmeden bağımsızdır.
  - **Eşik Değerlendirme Bileşeni:** Tahminin güvenini değerlendirir; tahmin motorundan bağımsız şekilde çalışabilir.
- **Karşılıdı Gereksinimler:**
  - FR-4: Model Çalıştırma

Bu ayristirma, modelin yüklenmesi, tahmin yapılması ve sonuç değerlendirmesinin birbirinden bağımsız olarak test edilmesine ve optimize edilmesine olanak sağlar.

## ResultPresentationModule

- **Kimlik:** ResultPresentationModule
- **Tür:** Kullanıcı arayüzü yazılım modülü
- **Amaç:** Model tarafından üretilen sonuçları kullanıcıya açık ve anlaşılır biçimde sunmak.
- **İşlev:** Tahmin sonucunu, güven oranını ve belirsizlik durumlarını görsel ve metinsel olarak göstermek; isteğe bağlı olarak sonuçları kaydetmek.
- **Altparçalar ve Ayristirma Gerekcesi:**
  - **Sonuç Görüntüleme Bileşeni:** Tahmin sonuçlarını gösterir; uyarılar ve kayıt yönetiminden bağımsızdır.
  - **Uyarı ve Mesaj Yönetimi Bileşeni:** Kullanıcıyı bilgilendirir; sonuç görüntülemeden bağımsızdır.
  - **Geçmiş Kayıt Yönetimi Bileşeni:** Sonuçları kaydeder ve geçmişe erişimi sağlar; diğer alt parçaların işlevini etkilemez.
- **Karşılıdı Gereksinimler:**
  - FR-5: Sonuç Gösterimi
  - FR-6: Eşik ve Uyarılar
  - FR-7: Sonuç Kaydı
  - FR-8: Hata Yönetimi

### 4.1.2. Veri Ayristirma

Bu altbölümde, yazılım sistemi içerisinde kullanılan veri elemanları, Bölüm 3'te tanımlanan tasarım varlıklarını temel alınarak ayristırılmıştır. Veri ayristırması, GŞÖA'da belirtilen gereksinimlerin karşılanabilmesi için gerekli veri türlerini, veri akışını ve veri dönüşümlerini açık biçimde ortaya koyacak şekilde yapılmıştır. Sistem içerisinde kullanılan başlıca veri elemanları aşağıda açıklanmıştır.

#### Ham Görüntü Verisi (RawImageData)

- **Tür:** Dijital görüntü verisi
- **Kaynak:** ImageAcquisitionModule

- **Kullanım Alanı:** ImagePreprocessingModule tarafından ön işleme girdi olarak kullanılır.
- **Biçim / Gösterim:** RGB formatında JPEG veya PNG dosyası.
- **İlk Değer:** Başlangıçta boş (null); kullanıcı tarafından görüntü alındığında oluşturulur.
- **Kabul Edilebilir Değerler:** Okunabilir, bozulmamış ve geçerli görüntü dosyaları
- **İlgili Gereksinimler:** FR-1 (Fotoğraf Çekme), FR-2 (Görüntü Yükleme)

### İşlenmiş Görüntü Verisi (PreprocessedImageData)

- **Tür:** Sayısal tensör verisi
- **Kaynak:** ImagePreprocessingModule
- **Kullanım Alanı:** ModelInferenceModule tarafından model girdisi olarak kullanılır.
- **Biçim / Gösterim:**  $224 \times 224 \times 3$  boyutunda normalize edilmiş tensör.
- **İlk Değer:** Ham görüntü alındığında oluşturulur
- **Kabul Edilebilir Değerler:** Modelin giriş katmanı tarafından kabul edilen boyut ve değer aralıkları.
- **İlgili Gereksinimler:** FR-3 (Görüntü İşleme)

### Model Tahmin Sonucu Verisi (PredictionResultData)

- **Tür:** Mantıksal ve sayısal veri bileşimi
- **Kaynak:** ModelInferenceModule
- **Kullanım Alanı:** ResultPresentationModule tarafından kullanıcıya sunulur ve isteğe bağlı olarak kaydedilir
- **Biçim / Gösterim:**
  - Sınıf etiketi: “Zehirli”, “Yenilebilir”, “Emin Değilim”
  - Güven oranı: %0–85
- **İlk Değer:** Model çalıştırılmadan önce tanımsızdır; tahmin sonucunda oluşturulur.
- **Kabul Edilebilir Değerler:** Etiketlerin belirtilen değerlerden biri olması ve güven oranının %0–85 aralığında olması
- **İlgili Gereksinimler:** FR-4 (Model Çalıştırma), FR-5 (Sonuç Gösterimi), FR-6 (Eşik ve Uyarılar)

### Sonuç Kayıt Verisi (ResultHistoryData)

- **Tür:** Yapılandırılmış kayıt verisi
- **Kaynak:** ResultPresentationModule
- **Kullanım Alanı:** Geçmiş analiz sonuçlarının görüntülenmesi ve yönetimi.
- **Biçim / Gösterim:** Yerel depolamada tutulan yapılandırılmış veri (ör. JSON)
- **İlk Değer:** Tahmin sonucu alındığında oluşturulur.
- **Kabul Edilebilir Değerler:** Geçerli tahmin sonucu ve güven oranı içeren kayıtlar.
- **İlgili Gereksinimler:** FR-7 (Sonuç Kaydı)

## 5. Bağımlılık Açıklaması

Bu bölümde, yazılım sisteminin oluşturulan tasarım varlıklarını arasındaki bağımlılıklar açıklanmaktadır. Bağımlılık açıklaması, sistem içindeki varlıkların çalışma sırası, veri paylaşımı, ilk değer atamaları ve kaynak kullanımları göz önünde bulundurularak yapılmıştır.

Bağımlilikler; sistemin doğru, güvenilir ve izlenebilir biçimde çalışabilmesi için tanımlanmıştır.

#### ImageAcquisitionModule Bağımlilikleri

- **Kimlik:** ImageAcquisitionModule
- **Tür:** Kullanıcı etkileşimli yazılım modülü
- **Amaç:**
- Görüntü alım sürecini başlatmak ve sisteme ham görüntü verisi sağlamak.
- **Bağımlilikler:**
  - Cihaz kamerası ve depolama birimi erişimi
  - İşletim sistemi izin yönetimi mekanizmaları
  - ResultPresentationModule ile kullanıcı yönlendirme ve hata bildirim etkileşimi

Bu modül, sistemin ilk çalışan bileşeni olup diğer modüller tarafından üretilen veriye bağımlı değildir. İlk değer ataması, kullanıcı etkileşimi ile gerçekleştirilir.

- **Kaynaklar:**
  - Mobil cihaz kamerası
  - Yerel depolama
  - İşletim sistemi API'leri

#### ImagePreprocessingModule Bağımlilikleri

- **Kimlik:** ImagePreprocessingModule
- **Tür:** Hesaplama odaklı yazılım modülü
- **Amaç:**
- Ham görüntü verisini modelin kabul edebileceği biçimde dönüştürmek.
- **Bağımlilikler:**
  - ImageAcquisitionModule tarafından üretilen ham görüntü verisi
  - Görüntü işleme kütüphaneleri

Bu modül, ImageAcquisitionModule çalışmadan başlatılamaz. İlk değer ataması, ham görüntü verisinin alınmasıyla gerçekleşir.

- **Kaynaklar:**
  - OpenCV kütüphanesi
  - CPU ve bellek kaynakları

#### ModelInferenceModule Bağımlilikleri

- **Kimlik:** ModelInferenceModule
- **Tür:** Model yürütme yazılım modülü
- **Amaç:**
- İşlenmiş görüntü üzerinde derin öğrenme modelini çalıştırarak tahmin üretmek.
- **Bağımlilikler:**
  - ImagePreprocessingModule tarafından üretilen işlenmiş görüntü verisi
  - TensorFlow Lite çalışma zamanı

- Model dosyasının cihazda yüklü olması

Bu modül, işlenmiş görüntü verisi olmadan çalışmaz. İlk değer ataması, modelin yüklenmesi ile yapılır.

- **Kaynaklar:**

- TensorFlow Lite kütüphanesi
- CPU / GPU / NPU (varsa)
- Bellek alanı

### ResultPresentationModule Bağımlılıkları

- **Kimlik:** ResultPresentationModule
- **Tür:** Kullanıcı arayüzü yazılım modülü
- **Amaç:**
- Tahmin sonuçlarını kullanıcıya sunmak ve sonuç yönetimini gerçekleştirmek.
- **Bağımlılıklar:**
  - ModelInferenceModule tarafından üretilen tahmin sonucu verisi
  - Kullanıcı etkileşimleri
  - Yerel depolama (opsiyonel sonuç kaydı için)

Bu modül, model tahmin sonucu olmadan anlamlı çıktı üretemez. İlk değer ataması, tahmin sonucu alındığında yapılır.

- **Kaynaklar:**

- React Native kullanıcı arayüzü bileşenleri
- Yerel depolama alanı
- İşletim sistemi UI servisleri

## 6. Arayüz Açıklaması

### ImageAcquisitionModule Arayüzü

- **Kullanıcı Etkileşimi:** Kamera aç, fotoğraf çek / galeri seç.
- **UI Elemanları:**
  - Fotoğraf çek butonu
  - Galeriden seç butonu
  - Önizleme ekranı (image view)
- **Varlıklarla Etkileşim:**
  - Kullanıcı eylemi → ImageAcquisitionModule
  - Ham görüntü verisi → Görüntü veri varlığı (ImageData)
  - Hata durumu → Hata varlığı (ErrorState)

### ImagePreprocessingModule Arayüzü

- **Kullanıcı Etkileşimi:** Yok, sistem içi çalışır.
- **UI Elemanları:** Arka planda işleme; kullanıcı görmez.
- **Varlıklarla Etkileşim:**
  - Ham görüntü verisi (ImageData) → ImagePreprocessingModule
  - İşlenmiş görüntü tensörü → Tensor varlığı (ProcessedImageTensor)
  - Hata durumu → Hata varlığı (ErrorState)

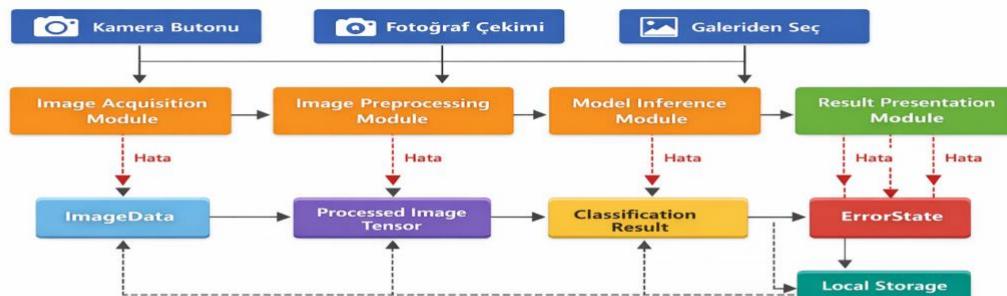
## ModelInferenceModule Arayüzü

- **Kullanıcı Etkileşimi:** Yok, sistem içi çalışır.
- **UI Elemanları:** Arka planda model çalıştırılır; kullanıcı sadece sonucu görür.
- **Varlıklarla Etkileşim:**
  - İşlenmiş görüntü tensörü (ProcessedImageTensor) → ModelInferenceModule
  - Tahmin sonucu (ClassificationResult) → ResultPresentationModule
  - Hata durumu → Hata varlığı (ErrorState)

## ResultPresentationModule Arayüzü

- **Kullanıcı Etkileşimi:** Tahmin sonuçlarını gösterir, uyarılar.
- **UI Elemanları:**
  - Sonuç metni (TextBox / Label)
  - Sonuç görseli (ImageView)
  - Uyarı mesajı (Popup / Toast)
  - Kaydet butonu (Opsiyonel)
- **Varlıklarla Etkileşim:**
  - Model tahmin sonucu (ClassificationResult) → ResultPresentationModule
  - Kullanıcı etkileşimi → ResultPresentationModule → Yerel kayıt varlığı (LocalStorage)

## Arayüz ve Varlıklar Arasındaki Etkileşim Şeması



## 7. Ayrıntılı Tasarım

Bu bölüm, yazılım sisteminin her bir tasarım varlığının iç yapısını ayrıntılı biçimde tanımlar. Tanımlar; gerçekleme aşamasında yazılım geliştiricilere yol gösterecek şekilde, işleme adımları ve kullanılan veri yapıları açıkça belirtilerek verilmiştir.

### ImageAcquisitionModule Ayrıntılı Tasarımı

#### Kimlik Belirleme

- **Varlık Adı:** ImageAcquisitionModule

#### İşleme

Bu modül kullanıcı etkileşimi ile çalışır ve aşağıdaki işlem adımlarını izler:

```
WAIT user_action
IF user_selects_camera THEN
    request_camera_permission
    capture_image
ELSE IF user_selects_gallery THEN
    request_storage_permission
    load_image
END IF

IF image_is_valid THEN
    output RawImageData
ELSE
    generate_error
END IF
```

Bu işlem sırasında kamera veya depolama izinleri işletim sistemi üzerinden kontrol edilir. Geçersiz veya okunamayan görüntüler hata olarak ele alınır.

#### Veri

- **Girdi:**
- Kullanıcı eylemi, görüntü dosyası (JPEG/PNG)
- **Çıktı:**
- RawImageData
- **Veri Yapısı:**
- RGB formatında bitmap / byte array

#### ImagePreprocessingModule Ayrıntılı Tasarımı

##### Kimlik Belirleme

- **Varlık Adı:** ImagePreprocessingModule

#### İşleme

Bu modül, ham görüntüyü modelin kabul edebileceği biçimde dönüştürür:

```
resize(image, 224x224)
convert_to_RGB(image)
normalize(pixel_values, range=0..1)
return PreprocessedImageData
```

Görüntü, sabit boyuta getirilir ve piksel değerleri normalize edilir.

## Veri

- **Girdi:**
- RawImageData
- **Çıktı:**
- PreprocessedImageData
- **Veri Yapısı:**
- Float32 türünde,  $224 \times 224 \times 3$  boyutlu tensör

## ModelInferenceModule Ayrıntılı Tasarımı

### Kimlik Belirleme

- **Varlık Adı:** ModelInferenceModule

### İşleme

Bu modül, TensorFlow Lite modeli kullanarak sınıflandırma işlemini gerçekleştirir:

```
load_tflite_model  
run_inference(PreprocessedImageData)  
calculate_confidence
```

```
IF confidence < threshold THEN
```

```
    label = "Emin Değilim"
```

```
ELSE
```

```
    label = predicted_class
```

```
END IF
```

```
return PredictionResultData
```

Belirlenen eşik değerinin altında kalan sonuçlar belirsiz olarak değerlendirilir.

## Veri

- **Girdi:**
- PreprocessedImageData
- **Çıktı:**
- PredictionResultData
- **Veri Yapısı:**
  - String: sınıf etiketi (“Zehirli”, “Yenilebilir”, “Emin Değilim”)
  - Float: güven oranı (%)

## ResultPresentationModule Ayrıntılı Tasarımı

### Kimlik Belirleme

- **Varlık Adı:** ResultPresentationModule

## İşleme

Bu modül, tahmin sonuçlarını kullanıcıya sunar ve isteğe bağlı kayıt işlemini yönetir:

```
display_label_and_confidence  
apply_color_coding
```

```
IF label == "Emin Değilim" THEN  
    show_warning_message  
END IF
```

```
IF user_requests_save THEN  
    store_result_locally  
END IF
```

Sonuçlar renk kodları ile görsel olarak desteklenir ve kullanıcı yönlendirilir.

## Veri

- **Girdi:**
- PredictionResultData
- **Çıktı:**
- Kullanıcı arayüzü çıktıları, opsiyonel kayıt verisi
- **Veri Yapısı:**
  - UI bileşenleri
  - Yerel depolamada tutulan JSON kayıtları