# 4.Düşük Seviyeli Tasarım

**1. Modüller ve Kütüphaneler**

**Açıklama:**

Proje için kullanılacak Python kütüphaneleri ve modülleri aşağıdaki gibidir:

* **OpenCV (cv2):** Görüntü işleme ve kameradan görüntü almak için kullanılır.
* **Mediapipe:** El hareketlerini tespit etmek ve taş-kağıt-makas hareketlerini tanımak için kullanılır.
* **NumPy:** Görüntü verilerini ve koordinatları işlemek için kullanılır.
* **Random:** Bilgisayarın rastgele bir taş, kağıt veya makas seçmesini sağlar.

**2. Sınıf ve Fonksiyon Tanımları**

**2.1 El Tespiti ve Tanıma**

* **El Tespiti Sınıfı (HandDetector)**
  + **Amaç:** Kamera görüntüsünden elin varlığını algılamak ve belirli hareketlerin (taş, kağıt, makas) tanınmasını sağlamak.
  + **Özellikler:**
    - mediapipe modelini başlatma.
    - El pozisyonlarını algılamak için fonksiyon.
  + **Metotlar:**
    - detect\_hands(image): Görüntüdeki elleri algılar.
    - classify\_hand(shape): Tespit edilen el hareketini sınıflandırır (taş, kağıt, makas).

**2.2 Oyun Mantığı**

* **Oyun Sınıfı (RockPaperScissors)**
  + **Amaç:** Oyuncunun seçimini algılamak, bilgisayarın seçim yapmasını sağlamak ve oyunun sonucunu hesaplamak.
  + **Özellikler:**
    - Oyuncunun seçimi (taş, kağıt, makas).
    - Bilgisayarın rastgele seçimi.
    - Skor takibi.
  + **Metotlar:**
    - get\_computer\_choice(): Bilgisayarın rastgele bir seçim yapmasını sağlar.
    - determine\_winner(player, computer): Oyuncu ve bilgisayar seçimlerine göre sonucu belirler.

**2.3 Kullanıcı Arayüzü**

* **Arayüz Fonksiyonları**
  + **Amaç:** Kullanıcıya kamera görüntüsünü ve oyun durumunu göstermek.
  + **Fonksiyonlar:**
    - draw\_text(image, text, position): Görüntü üzerine metin ekler.
    - display\_score(player\_score, computer\_score): Oyuncu ve bilgisayarın skorlarını ekrana yazar.
    - overlay\_image(image, choice): Bilgisayarın seçimini görsel olarak ekrana ekler.

**3. Veri Akışı**

**Açıklama:**

Oyun sırasında veri akışı şu şekilde gerçekleşir:

1. **Kamera Verisi Alımı:**
   * OpenCV aracılığıyla bilgisayarın kamerasından sürekli görüntü alınır.
2. **El Tespiti:**
   * Görüntü, HandDetector sınıfı kullanılarak analiz edilir ve oyuncunun el hareketi (taş, kağıt, makas) tanımlanır.
3. **Bilgisayar Seçimi:**
   * Bilgisayar rastgele bir seçim yapar.
4. **Sonuç Hesaplama:**
   * Oyuncunun ve bilgisayarın seçimleri determine\_winner() fonksiyonuna gönderilir ve kazanan belirlenir.
5. **Görüntüleme:**
   * Kamera görüntüsü, oyuncunun seçimi, bilgisayarın seçimi ve skorlar ekranda gösterilir.

**4. Oyun Akışı**

**Açıklama:**

Oyun sırasındaki adımlar detaylandırılır:

1. Kamera açılır ve kullanıcıdan hareket algılanır.
2. **Algılanan Hareketler:**
   * "Taş" için yumruk.
   * "Kağıt" için açık el.
   * "Makas" için V şeklindeki el işareti.
3. Bilgisayar rastgele seçim yapar.
4. Kazanan, oyunun kurallarına göre belirlenir:
   * Taş > Makas
   * Makas > Kağıt
   * Kağıt > Taş
5. Sonuç ve skor ekranda güncellenir.

**5. Veri Yapıları**

**Açıklama:**

Kullanılan veri yapıları ve formatlar:

* **Liste:** Bilgisayarın seçimleri için (choices = ['rock', 'paper', 'scissors']).
* **Sözlük:** Hareket sınıflandırması için ({'rock': 0, 'paper': 1, 'scissors': 2}).
* **Tam Sayı (int):** Oyuncu ve bilgisayarın skorları için.

**6. Algoritmalar**

**El Hareketi Sınıflandırma Algoritması:**

* Mediapipe koordinatları kullanarak:
  + Avuç içi genişliği, parmak uçlarının pozisyonları gibi ölçütlerle hareket tanımlanır.
  + Örnek: Eğer tüm parmaklar açıksa -> "Kağıt" olarak sınıflandırılır.

**Sonuç Belirleme Algoritması:**

* Oyuncu ve bilgisayar seçimleri karşılaştırılır:

**7. Performans ve Optimizasyon**

**Açıklama:**

* Kamera görüntüsü analizi, her döngüde gerçekleştirileceği için performans kritik bir unsurdur. OpenCV'nin optimize edilmiş metotları kullanılmalıdır.
* **FPS (Frames Per Second):** Algoritmanın gerçek zamanlı çalışması için döngü hızı optimize edilmelidir.
* Gereksiz görüntü işlemleri minimize edilerek kullanıcı deneyimi iyileştirilir.

**8. Test Senaryoları**

**Açıklama:**

* **Doğru Hareket Algılama:** Taş, kağıt, makas hareketleri doğru algılanıyor mu?
* **Kazanan Belirleme:** Oyuncu ve bilgisayarın seçimlerine göre doğru sonuç veriliyor mu?
* **Kullanıcı Arayüzü:** Görsel öğeler doğru görüntüleniyor mu?
* **Performans:** Program düşük ışık koşullarında da sorunsuz çalışıyor mu?