

```
1  import cv2
2  import numpy as np
3  import math
4  import csv
5  import time
6  from datetime import datetime
7
8  # Kameradan görüntü al
9  cap = cv2.VideoCapture(0)
10
11 # Referans noktası (değiştirilebilir)
12 ref_x, ref_y = 250, 250
13
14 # CSV dosya adı
15 dosya_adi = "angularinstant.csv"
16
17 # Kaydetme durumu ve zaman kontrolü
18 kayit_basladi = False
19 last_recorded_time = None
20
21 # Kullanıcı tarafından ayarlanabilir veri alma hızı (Hz)
22 interval_ms = 20
23
24 # Dosyayı başta sıfırlama
25 with open(dosya_adi, "w", newline="") as file:
26     writer = csv.writer(file)
27     writer.writerow(["Time", "Angle (deg)"]) # Başlıklar
28
29 while True:
30     ret, frame = cap.read()
31     if not ret:
32         break
33
34     # Görüntüyü HSV'ye çevir
35     hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
36
37     # Kırmızı renk aralıkları
38     lower_red1 = np.array([0, 120, 70]) # İlk aralık (0° civarı)
39     upper_red1 = np.array([10, 255, 255])
40     lower_red2 = np.array([170, 120, 70]) # İkinci aralık (360° civarı)
41     upper_red2 = np.array([180, 255, 255])
42
43     # Maskeler
44     mask1 = cv2.inRange(hsv, lower_red1, upper_red1)
45     mask2 = cv2.inRange(hsv, lower_red2, upper_red2)
46     mask = mask1 + mask2
47
48     # Gürültü azaltma
49     mask = cv2.medianBlur(mask, 5)
50
51     # Konturları bul
52     contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
53
54     # En büyük konturu bul
55     largest_contour = None
56     max_area = 0
57     for contour in contours:
58         area = cv2.contourArea(contour)
59         if area > max_area:
```

```

60         max_area = area
61         largest_contour = contour
62
63     ball_angle = None
64
65     if largest_contour is not None:
66         # Minimum çevreleyen daire
67         (x, y), radius = cv2.minEnclosingCircle(largest_contour)
68         if radius > 2: # Çok küçük nesneleri yok say
69             # Merkez noktayı ekrana yazdır
70             cv2.circle(frame, (int(x), int(y)), int(radius), (0, 255, 0), 2)
71             cv2.circle(frame, (int(x), int(y)), 5, (255, 0, 0), -1) # Merkez
noktası
72             cv2.putText(frame, f"Center: ({int(x)}, {int(y)})", (int(x) + 10, int(y)
- 10),
73                         cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 0), 2)
74
75             # Açı hesaplama
76             dx = x - ref_x
77             dy = ref_y - y # Pozitif y eksenini yukarı doğru
78             angle = math.degrees(math.atan2(dy, dx))
79             if angle <= 0:
80                 angle += 360 # Açıyı pozitif yap
81
82             ball_angle = angle
83             cv2.putText(frame, f"Angle: {int(angle)} deg", (int(x) + 10, int(y) +
20),
84                         cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 255, 255), 2)
85
86             # Kaydetme işlemi aktifse açıyı CSV'ye kaydet
87             if kayıt_basladi:
88                 current_time = datetime.now()
89                 formatted_time = current_time.strftime("%H:%M:%S.%f")[:-3] #
Milisaniyeye kadar
90
91                 # Eğer bir önceki kayıttan interval_ms geçmişse kaydet
92                 if last_recorded_time is None or (
93                     (current_time - last_recorded_time).total_seconds() * 1000
94                     >= interval_ms):
95                     with open(dosya_adi, "w", newline="") as file: # "w" ile tek
satır üzerine yazar
96                         writer = csv.writer(file)
97                         writer.writerow(["Time", "Angle (deg)"])
98                         writer.writerow([formatted_time, ball_angle])
99                         last_recorded_time = current_time
100
101             # Referans noktasını çiz
102             cv2.drawMarker(frame, (ref_x, ref_y), (0, 0, 255), cv2.MARKER_TILTED_CROSS, 20,
2)
103             cv2.putText(frame, f"Ref: ({ref_x}, {ref_y})", (ref_x + 10, ref_y - 10),
104                         cv2.FONT_HERSHEY_SIMPLEX, 0.6, (0, 0, 255), 2)
105
106             # Sonucu göster
107             cv2.imshow("Kamera", frame)
108             cv2.imshow("Maske", mask)
109
110             # Tuş kontrolü
111             key = cv2.waitKey(1) & 0xFF
112             if key == ord('q'): # Çıkış
113                 break
114             elif key == ord('w'): # Referans noktasını yukarı taşı

```

```
114         ref_y -= 5
115     elif key == ord('s'): # Referans noktasını aşağı taşı
116         ref_y += 5
117     elif key == ord('a'): # Referans noktasını sola taşı
118         ref_x -= 5
119     elif key == ord('d'): # Referans noktasını sağa taşı
120         ref_x += 5
121     elif key == ord('r'): # Kaydı başlat veya yeniden başlat
122         with open(dosya_adi, "w", newline="") as file:
123             writer = csv.writer(file)
124             writer.writerow(["Time", "Angle (deg)"]) # Başlıklar
125             kayit_basladi = True
126             last_recorded_time = None
127             print("Kayıt başladı ve dosya sıfırlandı.")
128     elif key == ord('t'): # Kaydı durdur
129         kayit_basladi = False
130         print("Kayıt durduruldu.")
131
132 cap.release()
133 cv2.destroyAllWindows()
134
```