



RAČUNALNI VID

MURSKO SREDIŠČE

2.7.2022.

Sadržaj

- Open CV
- Prepoznavanje pokreta rukom

OpenCV

- OpenCV je biblioteka za računalni vid
- Računalni vid - kako naučiti računalno da razumije što se nalazi na slici ili videu?
- Danas ćemo naučiti kako putem web kamere spojiti računalno s vanjskim svijetom i naučiti ga da razumije što vidi



Gdje se koristi OpenCV?

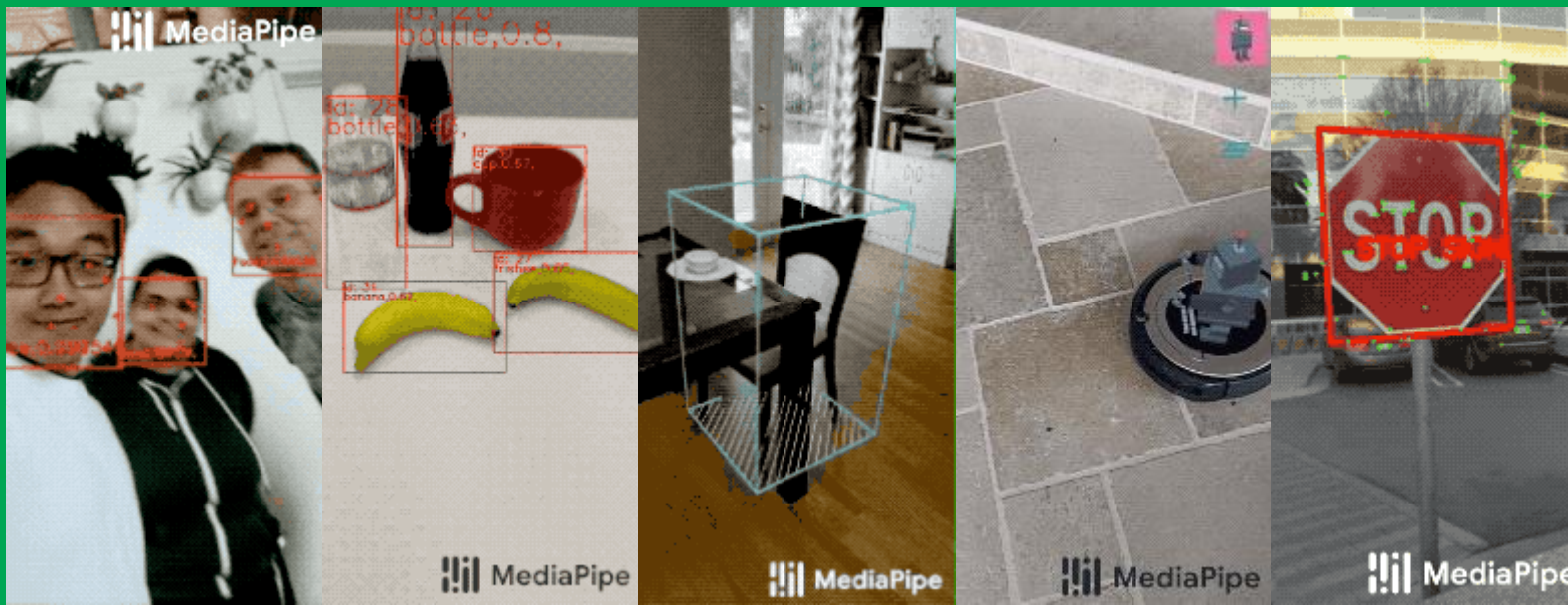
Primjeri primjene:

- Prepoznavanje objekata
- Prepoznavanje lica
- Prepoznavanje gesti
- Prepoznavanje pokreta
- Proširena stvarnost
- ...



Alati koje ćemo koristiti

- Python - programski jezik
- PyCharm - IDE
- OpenCV
- Mediapipe - ML biblioteka za medijski sadržaj



1. Kako prepoznati ruku?

- MediaPipe Hands
- Ključna obilježja ruke (eng. landmarks)



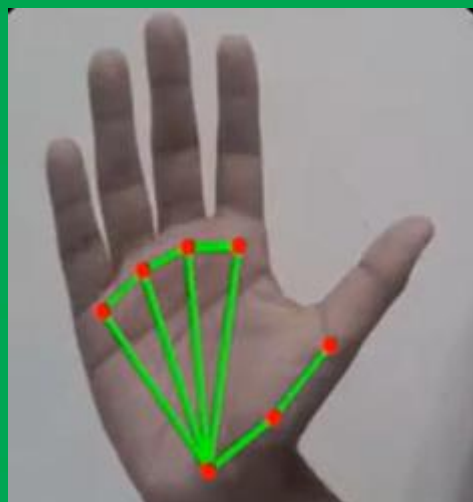
DONJA TOČKA DLANA



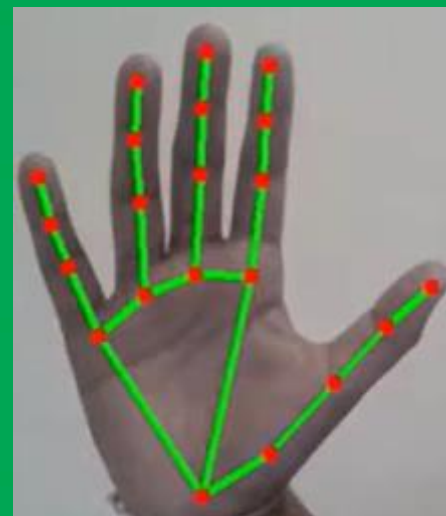
MOGUĆE PODRUČJE RUKE

1. Kako prepoznati ruku?

- Drugi korak je prepoznavanje zglobova



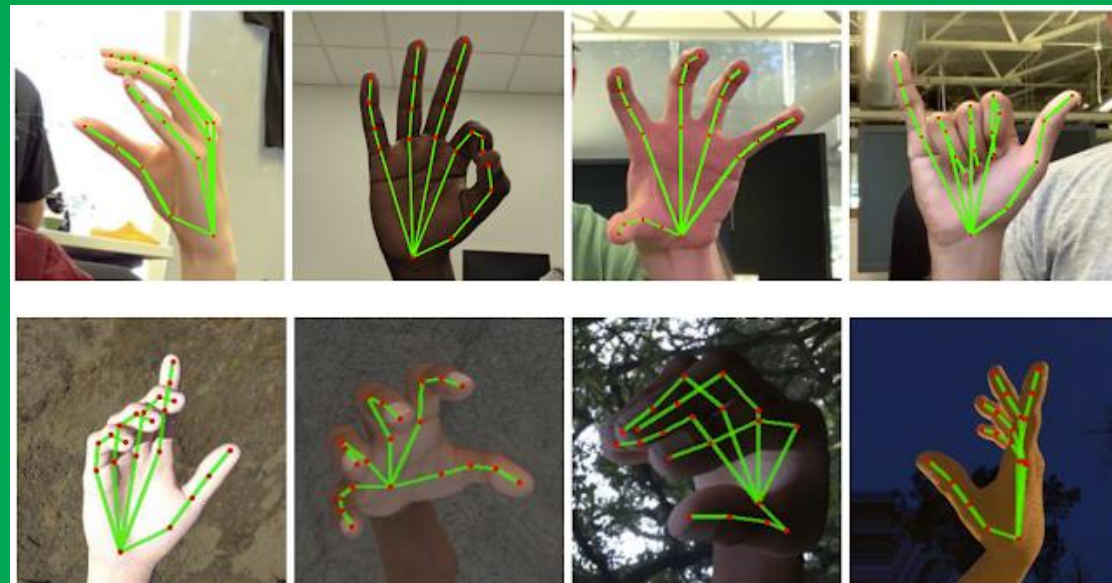
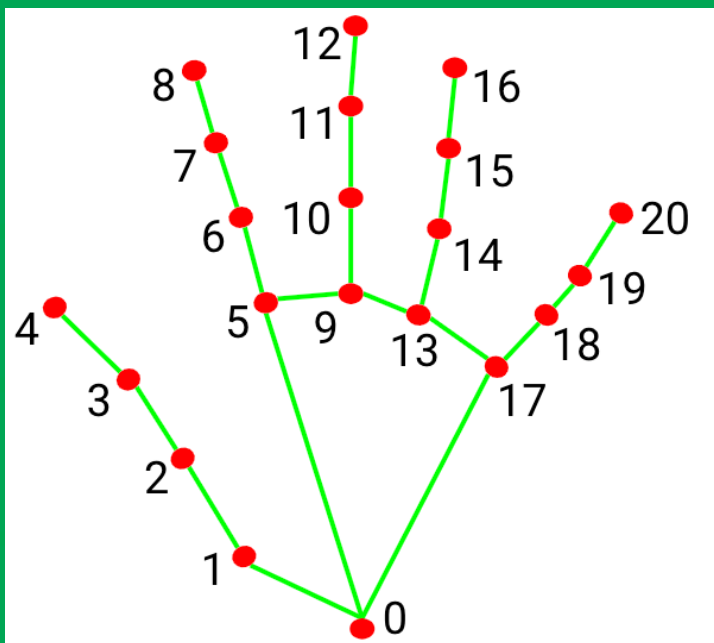
DETEKCIJA PRVIH ZGLOBOVA



GLAVNI ZGLOBOVI

1. Kako prepoznati ruku?

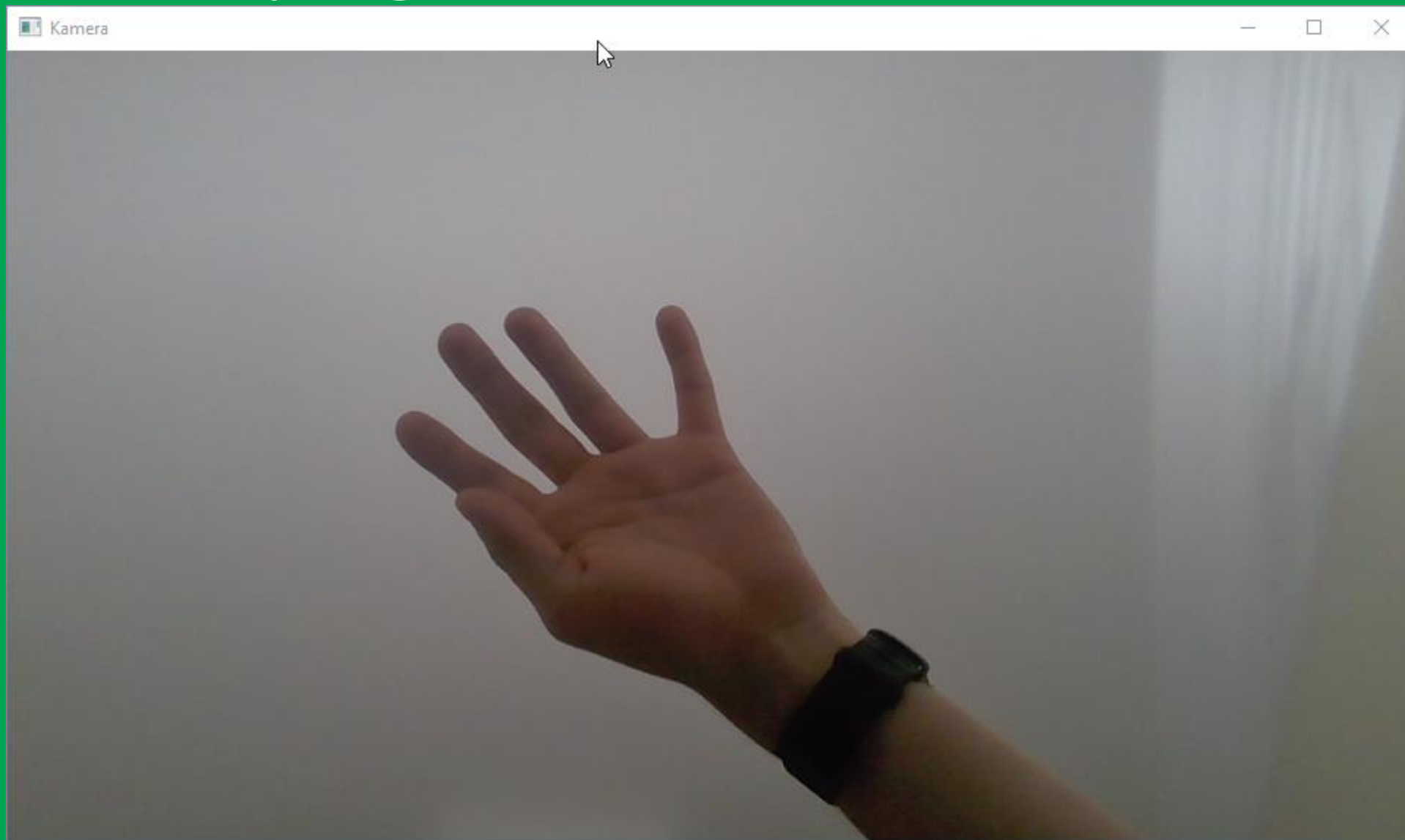
- Ukupno radimo sa 21 ključnom točkom
- Mediapipe će nam vratiti X i Y koordinatu svake točke na ekranu



2. Projekt

- <https://tinyurl.com/racunalni-vid>

Postavi program i video ulaz

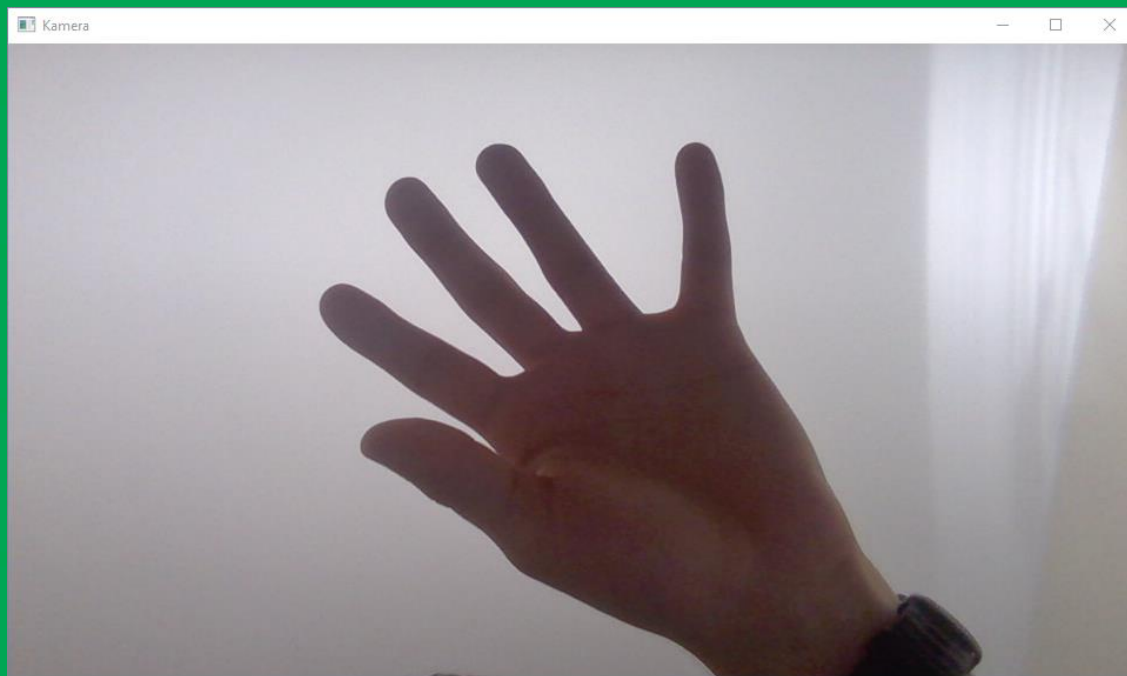


Postavi program i video ulaz



```
"""  
    Korak 1. Postavi program i video ulaz  
"""  
  
# ukljuci biblioteke  
import cv2  
  
# Postavi web kameru kao video ulaz i velicinu videa  
video = cv2.VideoCapture(0)  
video.set(3, 960)  
  
while True:  
    # Obradi ulazni video  
    status, slicica = video.read()  
  
    # Prikzi video na ekranu  
    cv2.imshow("Kamera", slicica)  
    # Sprijeci zatvaranje prozora  
    cv2.waitKey(1)
```

Postavi Mediapipe biblioteku



```
korak2 > <class 'mediapipe.python.solution_base.SolutionOutputs'>  
<class 'mediapipe.python.solution_base.SolutionOutputs'>  
<class 'mediapipe.python.solution_base.SolutionOutputs'>  
<class 'mediapipe.python.solution_base.SolutionOutputs'>  
<class 'mediapipe.python.solution_base.SolutionOutputs'>  
<class 'mediapipe.python.solution_base.SolutionOutputs'>  
<class 'mediapipe.python.solution_base.SolutionOutputs'>  
<class 'mediapipe.python.solution_base.SolutionOutputs'>
```

```
"""
    Korak 2. Postavi Mediapipe biblioteku
"""

# ukljuci biblioteke
import cv2
import mediapipe as mp

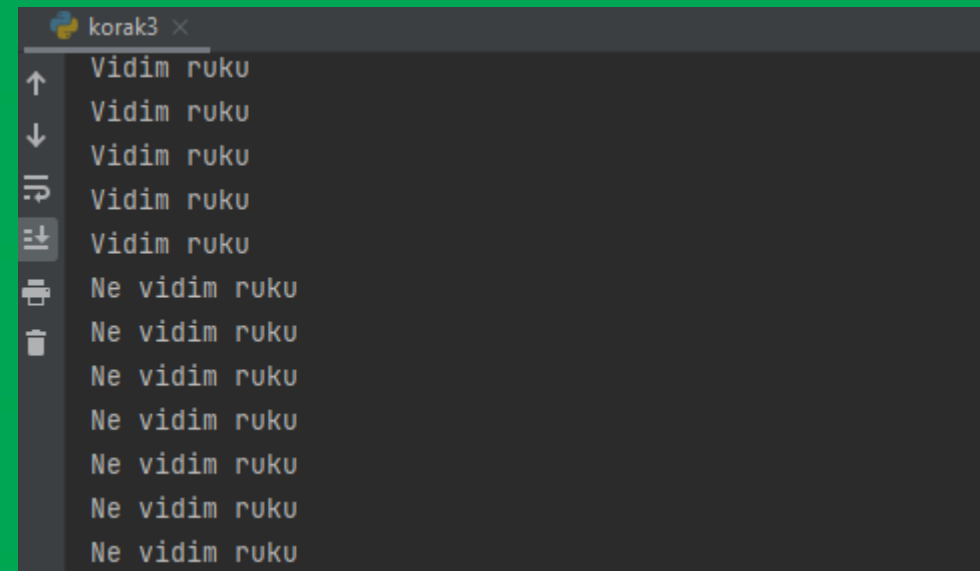
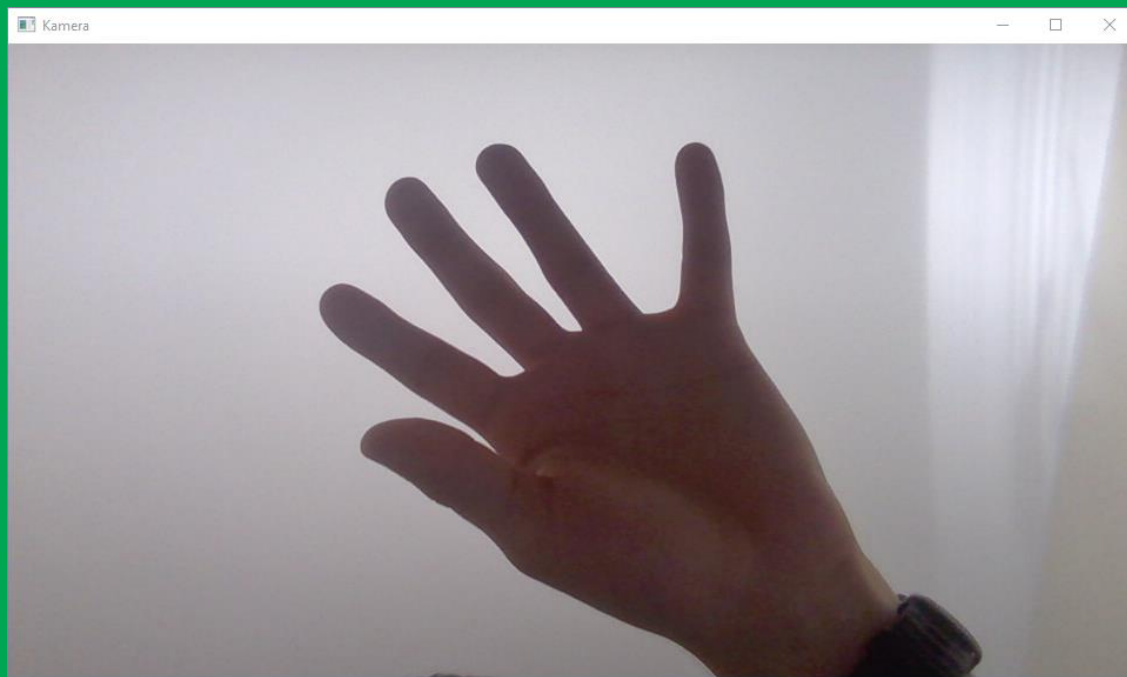
# Postavi web kameru kao video ulaz i velicinu videa
video = cv2.VideoCapture(0)
video.set(3, 960)

# Postavi mp hands algoritam
mp_hands = mp.solutions.hands
ruka = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.6)

while True:
    # Obradi ulazni video
    status, slicica = video.read()
    # Obradi ulazni video
    rezultat = ruka.process(slicica)
    # Ispisi rezultat u konzolu
    print(rezultat)

    # Prikzi video na ekranu
    cv2.imshow("Kamera", slicica)
    # Sprijeci zatvaranje prozora
    cv2.waitKey(1)
```

Pronađi ruku na kameri



```

"""
    Korak 3. Pronadi ruku
"""

# ukljuci biblioteke
import cv2
import mediapipe as mp

# Postavi web kameru kao video ulaz i velicinu videa
video = cv2.VideoCapture(0)
video.set(3, 960)

# Postavi mp hands algoritam
mp_hands = mp.solutions.hands
ruka = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.6)

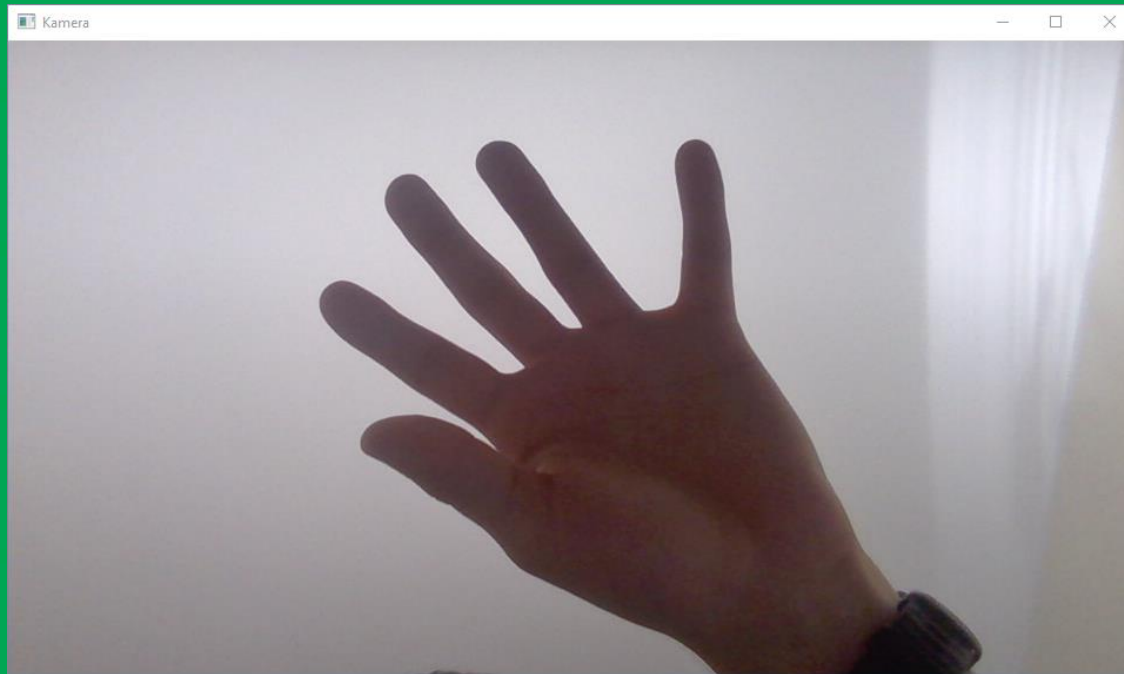
while True:
    # Obradi ulazni video
    status, slicica = video.read()
    rezultat = ruka.process(slicica)
    # Ispisi rezultat u konzolu
    # print(rezultat)

    # Ako je ruka nadena
    if rezultat.multi_hand_landmarks: # Ispisi ako je ruka nadena
        print("Vidim ruku")
    else:
        print("Ne vidim ruku")

    # Prikzi video na ekranu
    cv2.imshow("Kamera", slicica)
    # Sprijeci zatvaranje prozora
    cv2.waitKey(1)

```


Pronađi ključne točke ruke



```
korak4 x
↑
↓
↕
↔
✂
x: -0.08520849794149399
}
landmark {
  x: 0.6052727103233337
  y: 0.06418025493621826
  z: -0.10052017122507095
}
landmark {
  x: 0.7512387633323669
  y: 0.3889788091182709
  z: -0.04334348812699318
}
landmark {
  x: 0.7617449760437012
  y: 0.2602846622467041
  z: -0.07436232268810272
}
```



```
"""
    Korak 4. Pronadi ključne točke ruke
"""

# ukljuci biblioteke
import cv2
import mediapipe as mp

# Postavi web kameru kao video ulaz i velicinu videa
video = cv2.VideoCapture(0)
video.set(3, 960)

# Postavi mp hands algoritam
mp_hands = mp.solutions.hands
ruka = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.6)

while True:
    # Obradi ulazni video
    status, slicica = video.read()
    rezultat = ruka.process(slicica)
    # Ispisi rezultat u konzolu
    # print(rezultat)

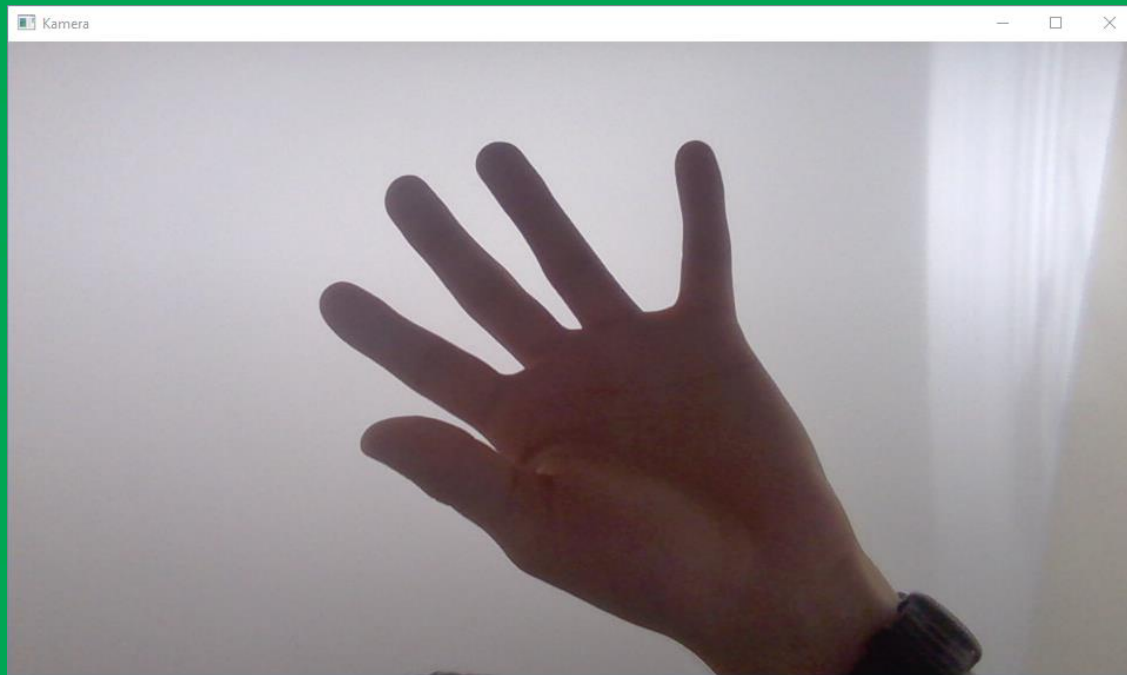
    # Ako je ruka nadena
    if rezultat.multi_hand_landmarks:
        # print("Vidim ruku")
        for obiljezje in rezultat.multi_hand_landmarks:
            print(obiljezje)

    # Prikzi video na ekranu
    cv2.imshow("Kamera", slicica)
    # Sprijeci zatvaranje prozora
    cv2.waitKey(1)
```



```
# Ako je roka nadena
if rezultat.multi_hand_landmarks:
    # print("Vidim ruku")
    for obiljezje in rezultat.multi_hand_landmarks:
        print(obiljezje)
```

Izračunaj poziciju točaka



↑	15	581	133
↓	16	561	92
↶	17	681	254
↷	18	696	192
↵	19	707	149
↴	20	715	108

```

"""
    Korak 5. Izracunaj poziciju tocaka
"""

# ukljuci biblioteke
import cv2
import mediapipe as mp

# Postavi web kameru kao video ulaz i velicinu videa
video = cv2.VideoCapture(0)
video.set(3, 960)

# Postavi mp hands algoritam
mp_hands = mp.solutions.hands
ruka = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.6)

while True:
    # Obradi ulazni video
    status, slicica = video.read()
    rezultat = ruka.process(slicica)
    # Ispisi rezultat u konzolu
    # print(rezultat)

    # Ako je ruka nadena
    if rezultat.multi_hand_landmarks:
        # print("Vidim ruku")
        for obiljezje in rezultat.multi_hand_landmarks:
            #print(obiljezje)
            for id, o in enumerate(obiljezje.landmark):
                # Izracunaj stvarnu poziciju na ekranu
                visina, sirina, dubina = slicica.shape
                x = int(o.x * sirina)
                y = int(o.y * visina)
                print(id, x, y)

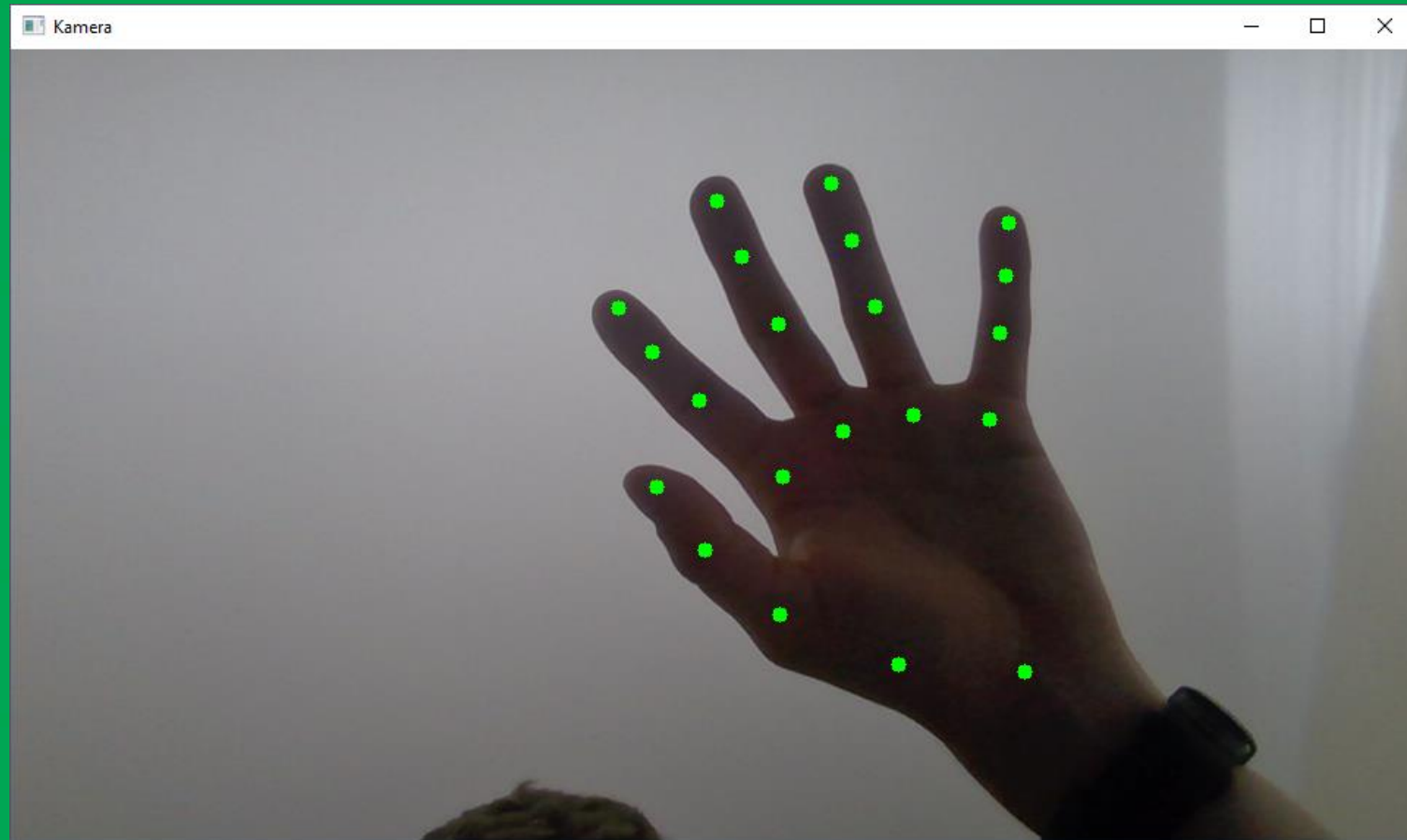
    # Prikzi video na ekranu
    cv2.imshow("Kamera", slicica)
    # Sprijeci zatvaranje prozora
    cv2.waitKey(1)

```



```
# Ako je ruka nadena
if rezultat.multi_hand_landmarks:
    # print("Vidim ruku")
    for obiljezje in rezultat.multi_hand_landmarks:
        #print(obiljezje)
        for id, o in enumerate(obiljezje.landmark):
            # Izracunaj stvarnu poziciju na ekranu
            visina, sirina, dubina = slicica.shape
            x = int(o.x * sirina)
            y = int(o.y * visina)
            print(id, x, y)
```

Nacrtaj točke na ekranu




```

"""
    Korak 6. Nacrtaј tocke
"""

# ukljuci biblioteke
import cv2
import mediapipe as mp

# Postavi web kameru kao video ulaz i velicinu videa
video = cv2.VideoCapture(0)
video.set(3, 960)

# Postavi mp hands algoritam
mp_hands = mp.solutions.hands
ruka = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.6)

# Boje
zeleno = (0, 255, 0)

while True:
    # Obradi ulazni video
    status, slicica = video.read()
    rezultat = ruka.process(slicica)
    # Ispisi rezultat u konzolu
    # print(rezultat)

    # Ako je ruka nadena
    if rezultat.multi_hand_landmarks:
        # print("Vidim ruku")
        for obiljezje in rezultat.multi_hand_landmarks:
            # print(obiljezje)
            for id, o in enumerate(obiljezje.landmark):
                # Izracunaj stvarnu poziciju na ekranu
                visina, sirina, dubina = slicica.shape
                x = int(o.x * sirina)
                y = int(o.y * visina)
                print(id, x, y)
                cv2.circle(slicica, (x,y), 5, zeleno, cv2.FILLED)

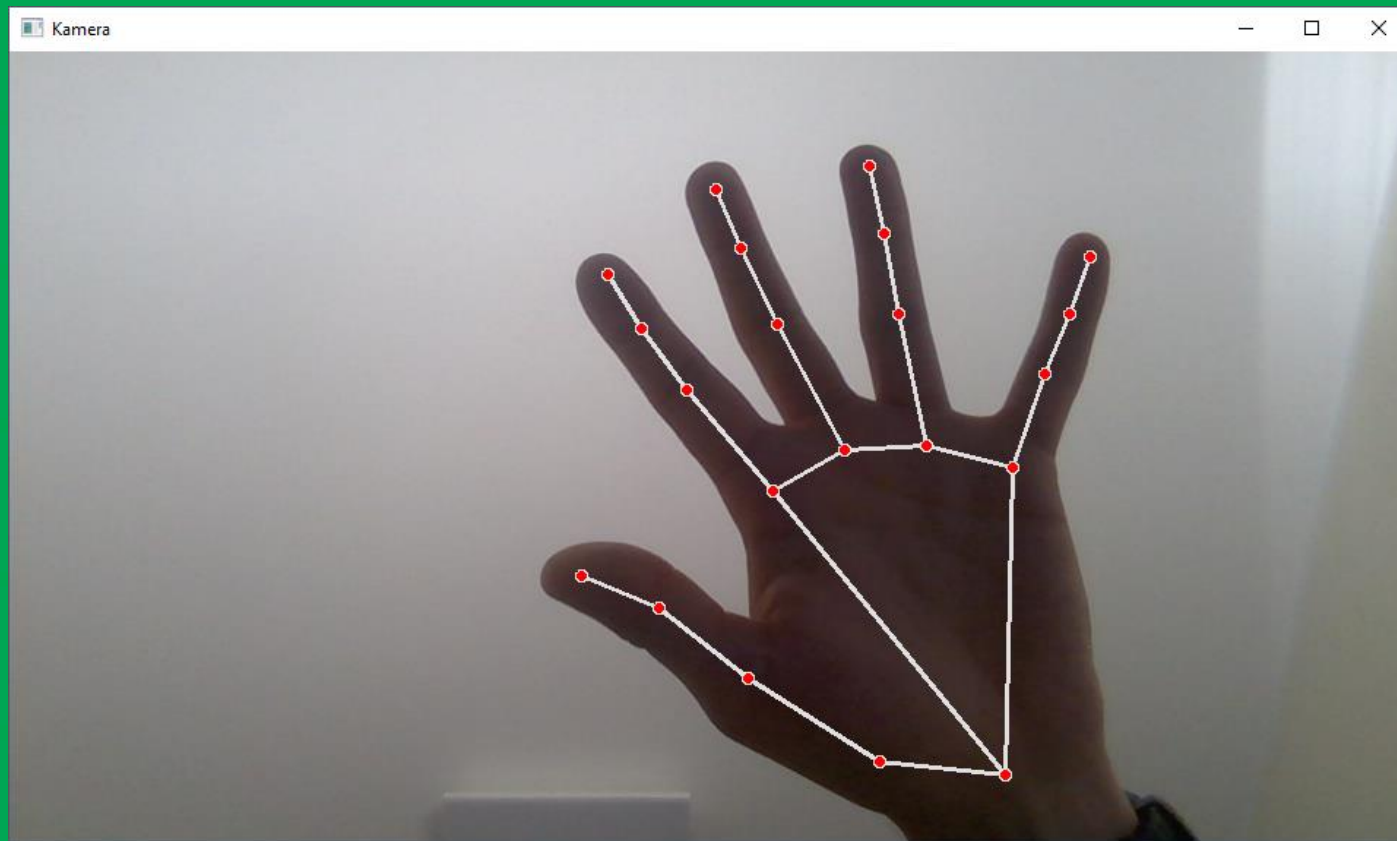
    # Prikazi video na ekranu
    cv2.imshow("Kamera", slicica)
    # Sprijeci zatvaranje prozora
    cv2.waitKey(1)

```



```
# Ako je ruka nadena
if rezultat.multi_hand_landmarks:
    # print("Vidim ruku")
    for obiljezje in rezultat.multi_hand_landmarks:
        # print(obiljezje)
        for id, o in enumerate(obiljezje.landmark):
            # Izracunaj stvarnu poziciju na ekranu
            visina, sirina, dubina = slicica.shape
            x = int(o.x * sirina)
            y = int(o.y * visina)
            print(id, x, y)
            cv2.circle(slicica, (x,y), 5, zeleno, cv2.FILLED)
```

Nacrtaj sve karakteristike na ekranu



```

"""
    Korak 7. Crtanje ključnih karakteristika
"""

# ukljuci biblioteke
import cv2
import mediapipe as mp

# Postavi web kameru kao video ulaz i velicinu videa
video = cv2.VideoCapture(0)
video.set(3, 960)

# Postavi mp hands algoritam
mp_hands = mp.solutions.hands
ruka = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.6)

# Postavi pomocne funkcije za crtanje
mp_crtanje = mp.solutions.drawing_utils

# Boje
zeleno = (0, 255, 0)

while True:
    # Obradi ulazni video
    status, slicica = video.read()
    rezultat = ruka.process(slicica)
    # Ispisi rezultat u konzolu
    # print(rezultat)

    # Ako je ruka nadena
    if rezultat.multi_hand_landmarks:
        # print("Vidim ruku")
        for obiljezje in rezultat.multi_hand_landmarks:
            # print(obiljezje)
            for id, o in enumerate(obiljezje.landmark):
                # Izracunaj stvarnu poziciju na ekranu
                visina, sirina, dubina = slicica.shape
                x = int(o.x * sirina)
                y = int(o.y * visina)
                print(id, x, y)
                # cv2.circle(slicica, (x,y), 5, zeleno, cv2.FILLED)
                mp_crtanje.draw_landmarks(slicica, obiljezje, mp_hands.HAND_CONNECTIONS)

    # Prikazi video na ekranu
    cv2.imshow("Kamera", slicica)
    # Sprijeci zatvaranje prozora
    cv2.waitKey(1)

```

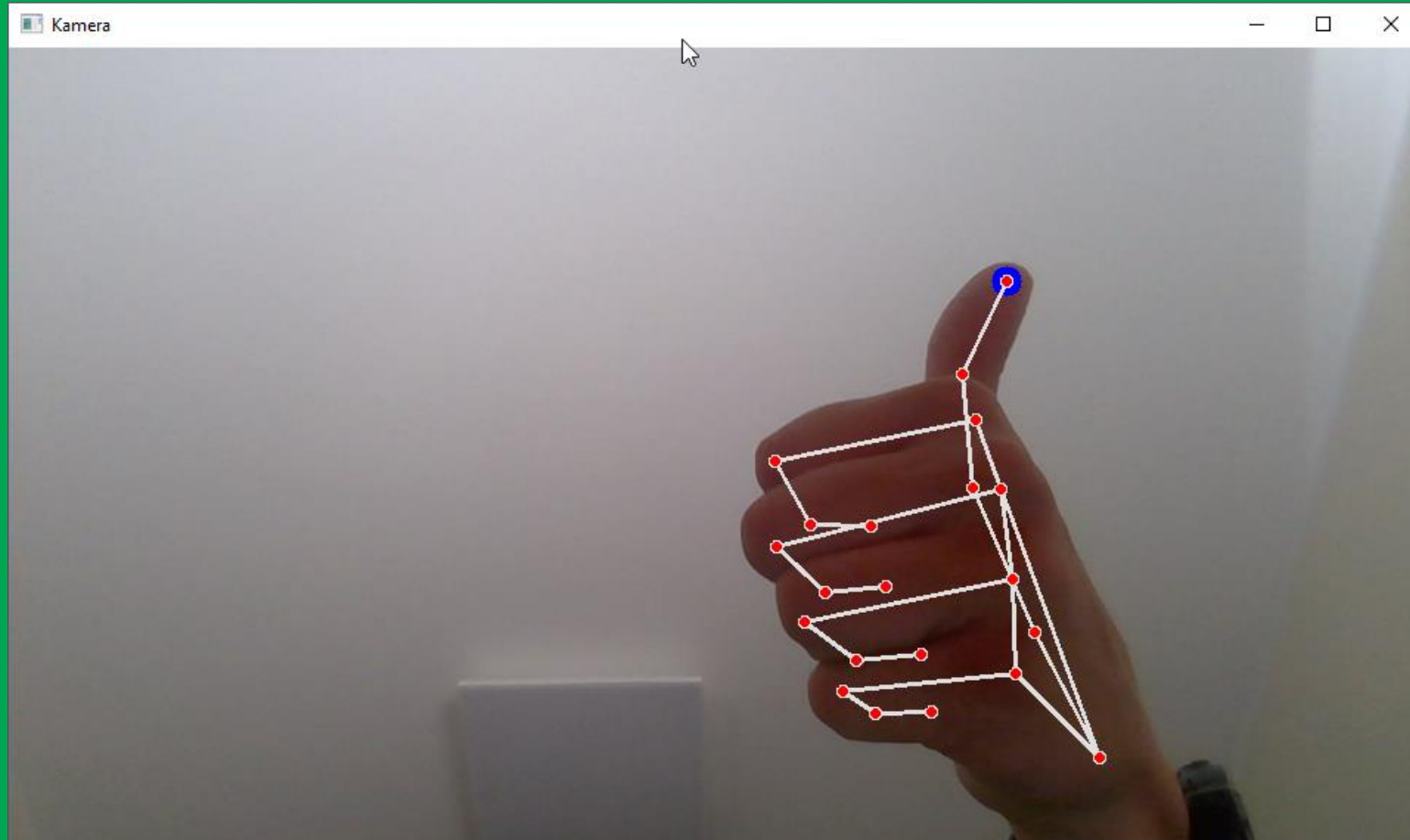


```
# Postavi pomocne funkcije za crtanje
mp_crtanje = mp.solutions.drawing_utils

...

# Ako je ruka nadena
if rezultat.multi_hand_landmarks:
    # print("Vidim ruku")
    for obiljezje in rezultat.multi_hand_landmarks:
        #print(obiljezje)
        for id, o in enumerate(obiljezje.landmark):
            # Izracunaj stvarnu poziciju na ekranu
            visina, sirina, dubina = slicica.shape
            x = int(o.x * sirina)
            y = int(o.y * visina)
            print(id, x, y)
            # cv2.circle(slicica, (x,y), 5, zeleno, cv2.FILLED)
        mp_crtanje.draw_landmarks(slicica, obiljezje, mp_hands.HAND_CONNECTIONS)
```

Označi vrh palca



```

"""
    Korak 8. Oznaci vrh palca
"""

# ukljuci biblioteke
import cv2
import mediapipe as mp

# Postavi web kameru kao video ulaz i velicinu videa
video = cv2.VideoCapture(0)
video.set(3, 960)

# Postavi mp hands algoritam
mp_hands = mp.solutions.hands
ruka = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.6)

# Postavi pomocne funkcije za crtanje
mp_crtanje = mp.solutions.drawing_utils

# Boje
zeleno = (0, 255, 0)
plavo = (255, 0, 0)

while True:
    # Obradi ulazni video
    status, slicica = video.read()
    rezultat = ruka.process(slicica)
    # Ispisi rezultat u konzolu
    # print(rezultat)

    # Ako je ruka nadena
    if rezultat.multi_hand_landmarks:
        # print("Vidim ruku")
        for obiljezje in rezultat.multi_hand_landmarks:
            # print(obiljezje)
            for id, o in enumerate(obiljezje.landmark):
                # Izracunaj stvarnu poziciju na ekranu
                visina, sirina, dubina = slicica.shape
                x = int(o.x * sirina)
                y = int(o.y * visina)
                print(id, x, y)

                # vrh palca
                if(id == 4):
                    palac_x = x
                    palac_y = y
                    cv2.circle(slicica, (palac_x, palac_y), 10, plavo, cv2.FILLED)

                #cv2.circle(slicica, (x,y), 5, zeleno, cv2.FILLED)
                mp_crtanje.draw_landmarks(slicica, obiljezje, mp_hands.HAND_CONNECTIONS)

    # Prikazi video na ekranu
    cv2.imshow("Kamera", slicica)
    # Sprijeci zatvaranje prozora
    cv2.waitKey(1)

```

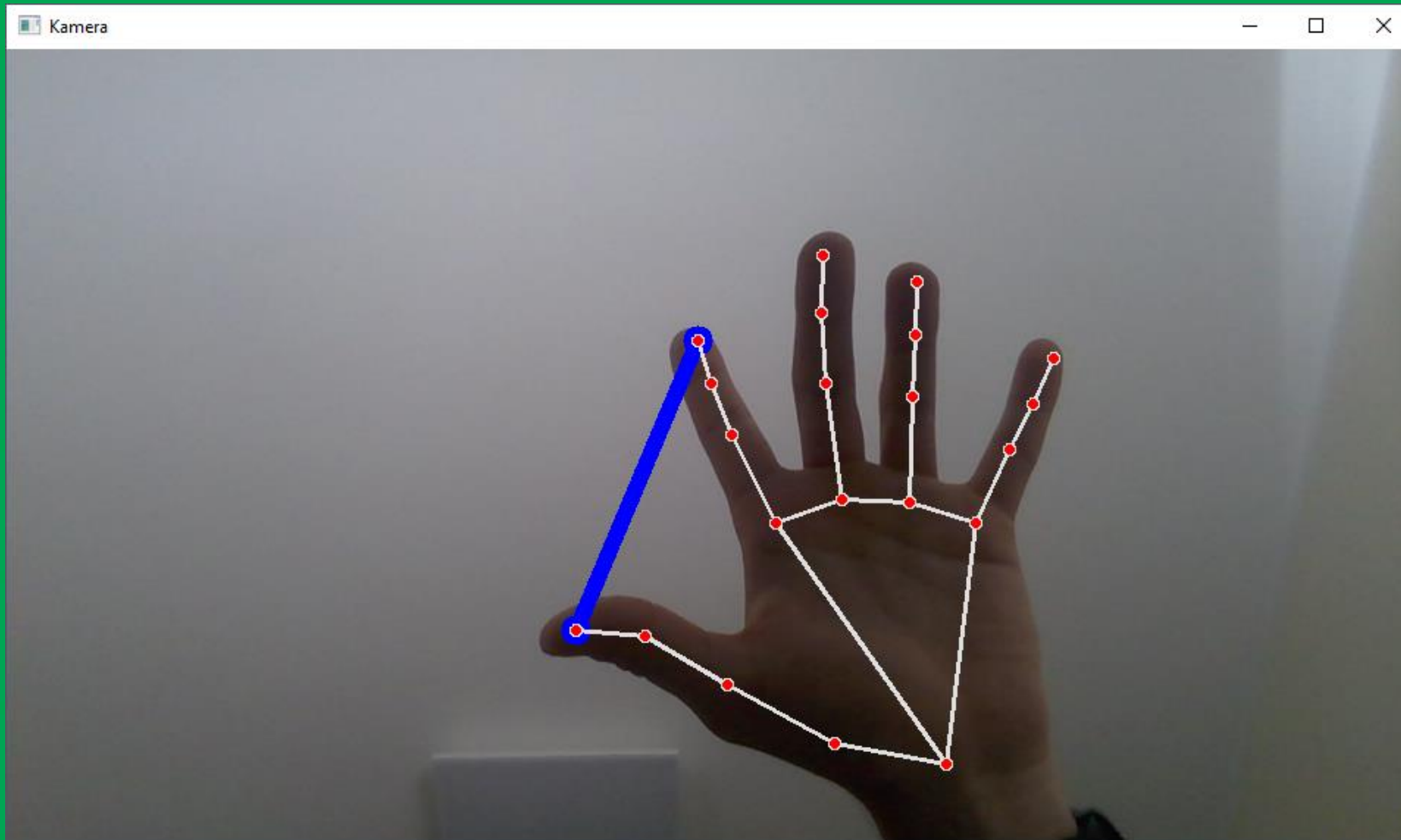



```
if rezultat.multi_hand_landmarks:
    # print("Vidim ruku")
    for obiljezje in rezultat.multi_hand_landmarks:
        #print(obiljezje)
        for id, o in enumerate(obiljezje.landmark):
            # Izracunaj stvarnu poziciju na ekranu
            visina, sirina, dubina = slicica.shape
            x = int(o.x * sirina)
            y = int(o.y * visina)
            print(id, x, y)

        # vrh palca
        if(id == 4):
            palac_x = x
            palac_y = y
            cv2.circle(slicica, (palac_x, palac_y), 10, plavo, cv2.FILLED)

        #cv2.circle(slicica, (x,y), 5, zeleno, cv2.FILLED)
        mp_crtanje.draw_landmarks(slicica, obiljezje, mp_hands.HAND_CONNECTIONS)
```

Spoji vrh palca i kažiprst



```

"""
    Korak 9. Spoji palac i kaziprst
"""
# ukljuci biblioteke
import cv2
import mediapipe as mp

# Postavi web kameru kao video ulaz i velicinu videa
video = cv2.VideoCapture(0)
video.set(3, 960)

# Postavi mp hands algoritam
mp_hands = mp.solutions.hands
ruka = mp_hands.Hands(
    max_num_hands=1,
    min_detection_confidence=0.7,
    min_tracking_confidence=0.6)

# Postavi pomocne funkcije za crtanje
mp_crtanje = mp.solutions.drawing_utils

# Boje
zeleno = (0, 255, 0)
plavo = (255, 0, 0)

while True:
    # Obradi ulazni video
    status, slicica = video.read()
    rezultat = ruka.process(slicica)
    # Ispisi rezultat u konzolu
    # print(rezultat)

    # Ako je ruka nadena
    if rezultat.multi_hand_landmarks:
        # print("Vidim ruku")
        for obiljezje in rezultat.multi_hand_landmarks:
            # print(obiljezje)
            for id, o in enumerate(obiljezje.landmark):
                # Izracunaj stvarnu poziciju na ekranu
                visina, sirina, dubina = slicica.shape
                x = int(o.x * sirina)
                y = int(o.y * visina)
                print(id, x, y)
                # vrh palca
                if id == 4:
                    palac_x = x
                    palac_y = y
                    cv2.circle(slicica, (palac_x, palac_y), 10, plavo, cv2.FILLED)
                # vrh kaziprsta
                if id == 8:
                    kaziprst_x = x
                    kaziprst_y = y
                    cv2.circle(slicica, (kaziprst_x, kaziprst_y), 10, plavo, cv2.FILLED)
                # spoji palac i kaziprst
                cv2.line(slicica, (palac_x, palac_y), (kaziprst_x, kaziprst_y), plavo, 10)
            #cv2.circle(slicica, (x,y), 5, zeleno, cv2.FILLED)
            mp_crtanje.draw_landmarks(slicica, obiljezje, mp_hands.HAND_CONNECTIONS)

    # Prikazi video na ekranu
    cv2.imshow("Kamera", slicica)
    # Sprijeci zatvaranje prozora
    cv2.waitKey(1)

```

STEMAJMO



Projekt je sufinancirala Europska unija iz Europskog socijalnog fonda. Projekt sufinancira Ured za udruge Vlade Republike Hrvatske. Sadržaj prezentacije isključiva je odgovornost Centra tehničke kulture Rijeka.